REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704-0188

PLEASE DO NOT RETURN YOUR FORM TO THE ABOVE ORGANIZATION.

1. REPORT DATE *(DD-MM-YYYY)*	2. REPORT TYPE	3. DATES COVERED *(From - To)*
24-06-2013	Final	27-03-2009 - 05-04-2013

4. TITLE AND SUBTITLE
A Posteriori Error Analysis and Uncertainty Quantification for Adaptive Multiscale Operator Decomposition Methods for Multiphysics Problems

5a. CONTRACT NUMBER

5b. GRANT NUMBER
HDTRA1-09-0036

5c. PROGRAM ELEMENT NUMBER

6. AUTHOR(S)
Estep, Donald J.
Holst, Michael

5d. PROJECT NUMBER

5e. TASK NUMBER

5f. WORK UNIT NUMBER

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)
Department of Statistics, Colorado State University, Fort Collins, CO 80523
Department of Mathematics, University of California San Diego, La Jolla, CA 92093

8. PERFORMING ORGANIZATION REPORT NUMBER

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)
Defense Threat Reduction Agency
8725 John J Kingman Rd
Stop 6201
Ft Belvoir, VA 22060

10. SPONSOR/MONITOR'S ACRONYM(S)

11. SPONSOR/MONITOR'S REPORT NUMBER(S)

12. DISTRIBUTION/AVAILABILITY STATEMENT
Distro A

13. SUPPLEMENTARY NOTES

14. ABSTRACT

This project was concerned with numerical solution and quantification of accuracy of numerical solution of multiphysics systems that couple together different physical processes acting across a large range of scales are encountered in virtually all areas of interest to Defense Threat Reduction Agency. The overall goal of this project was to develop a mathematically sound yet computationally practical methodology for estimating and mitigating the effects of error and uncertainty in information computed from numerical solutions of multiphysics problems. In addition, the proposed work provided computational tools for model prediction, uncertainty quantification, parameter optimization, and verification and validation. A primary goal of this project was to construct an a posteriori analysis framework for a number of multiphysics problems important to the DTRA mission. They used the results to devise innovative adaptive discretization algorithms. They also used the methods to develop and analyze fast methods for forward and inverse sensitivity analysis of multiphysics problems.

15. SUBJECT TERMS
multiphysics systems, operator decomposition, a posteriori error analysis, differential equations on manifolds, adaptive error control, convergence analysis, stochastic sensitivity analysis

16. SECURITY CLASSIFICATION OF:			17. LIMITATION OF ABSTRACT	18. NUMBER OF PAGES	19a. NAME OF RESPONSIBLE PERSON
a. REPORT	b. ABSTRACT	c. THIS PAGE			19b. TELEPHONE NUMBER *(Include area code)*
U	U	U	UU	278	

Reset

INSTRUCTIONS FOR COMPLETING SF 298

1. REPORT DATE. Full publication date, including day, month, if available. Must cite at least the year and be Year 2000 compliant, e.g. 30-06-1998; xx-06-1998; xx-xx-1998.

2. REPORT TYPE. State the type of report, such as final, technical, interim, memorandum, master's thesis, progress, quarterly, research, special, group study, etc.

3. DATES COVERED. Indicate the time during which the work was performed and the report was written, e.g., Jun 1997 - Jun 1998; 1-10 Jun 1996; May - Nov 1998; Nov 1998.

4. TITLE. Enter title and subtitle with volume number and part number, if applicable. On classified documents, enter the title classification in parentheses.

5a. CONTRACT NUMBER. Enter all contract numbers as they appear in the report, e.g. F33615-86-C-5169.

5b. GRANT NUMBER. Enter all grant numbers as they appear in the report, e.g. AFOSR-82-1234.

5c. PROGRAM ELEMENT NUMBER. Enter all program element numbers as they appear in the report, e.g. 61101A.

5d. PROJECT NUMBER. Enter all project numbers as they appear in the report, e.g. 1F665702D1257; ILIR.

5e. TASK NUMBER. Enter all task numbers as they appear in the report, e.g. 05; RF0330201; T4112.

5f. WORK UNIT NUMBER. Enter all work unit numbers as they appear in the report, e.g. 001; AFAPL30480105.

6. AUTHOR(S). Enter name(s) of person(s) responsible for writing the report, performing the research, or credited with the content of the report. The form of entry is the last name, first name, middle initial, and additional qualifiers separated by commas, e.g. Smith, Richard, J, Jr.

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES). Self-explanatory.

8. PERFORMING ORGANIZATION REPORT NUMBER. Enter all unique alphanumeric report numbers assigned by the performing organization, e.g. BRL-1234; AFWL-TR-85-4017-Vol-21-PT-2.

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES). Enter the name and address of the organization(s) financially responsible for and monitoring the work.

10. SPONSOR/MONITOR'S ACRONYM(S). Enter, if available, e.g. BRL, ARDEC, NADC.

11. SPONSOR/MONITOR'S REPORT NUMBER(S). Enter report number as assigned by the sponsoring/ monitoring agency, if available, e.g. BRL-TR-829; -215.

12. DISTRIBUTION/AVAILABILITY STATEMENT. Use agency-mandated availability statements to indicate the public availability or distribution limitations of the report. If additional limitations/ restrictions or special markings are indicated, follow agency authorization procedures, e.g. RD/FRD, PROPIN, ITAR, etc. Include copyright information.

13. SUPPLEMENTARY NOTES. Enter information not included elsewhere such as: prepared in cooperation with; translation of; report supersedes; old edition number, etc.

14. ABSTRACT. A brief (approximately 200 words) factual summary of the most significant information.

15. SUBJECT TERMS. Key words or phrases identifying major concepts in the report.

16. SECURITY CLASSIFICATION. Enter security classification in accordance with security classification regulations, e.g. U, C, S, etc. If this form contains classified information, stamp classification level on the top and bottom of this page.

17. LIMITATION OF ABSTRACT. This block must be completed to assign a distribution limitation to the abstract. Enter UU (Unclassified Unlimited) or SAR (Same as Report). An entry in this block is necessary if the abstract is to be limited.

Contents

Defense Threat Reduction Agency
Final Report

DTRA Grant: HDTRA1-09-0036

Title: *A Posteriori Error Analysis and Uncertainty Quantification for Adaptive Multiscale Operator Decomposition Methods for Multiphysics Problems*

Project Start Date: March 27, 2009
Project End Date: April 5, 2013

Principal Investigator
Donald Estep
Department of Statistics
Department of Mathematics
Colorado State University
Fort Collins, CO 80523
estep@stat.colostate.edu

Co-Principal Investigator
Michael Holst
Department of Mathematics
Department of Physics
University of California, San Diego
La Jolla, CA 92093
mholst@math.ucsd.edu

1. Accomplishments

1.1. Summary of accomplishments

The mission of the Defense Threat Reduction Agency requires the quantitative study and accurate prediction for complex multiphysics systems that couple together physical processes spanning wide range of scales in behavior. Treatment of such systems depends on accurate numerical simulation of mathematical models expressed as systems of partial differential equations posed on domains with complicated geometry. Prediction of the behavior involves treating the propagation of stochastic uncertainty through the mathematical models and solving inverse problems for determining parameters based on observations on model output. Quantifying the accuracy of such computations requires accurate estimation of the numerical error in quantities of interest computed from numerical solutions that take into account all sources of error, e.g. from discretization, representation of geometry, finite sampling.

This project focuses on development of mathematical tools for dealing with these problems in the context of multiphysics models of interest using relevant numerical methods to the mission of the DTRA. The main approach is a posteriori error analysis based on computable residuals, solution of adjoint problems, and variational analysis. This approach estimates the error in specified quantities of interest. Computable residuals involving the approximate solution are used to quantify the size of various discretization errors while the solution of adjoint equations (generalized Green's functions) are used to quantify the effects of stability in producing errors. Much of the project dealt with dealing the significant mathematical issues that arise when numerically solving complex multiphysics models. Practical computational constraints requires the use of a wide variety of discretization approaches, e.g. operator decomposition and splitting, explicit time integration, iterative solution methods with few iterations, finite volume and specialized finite difference methods. The introduction of such techniques complicates both the identification of suitable residuals and definition of suitable adjoint problems. The project also dealt with issues arising in "multi-discretization" approaches, when various components of a coupled system are solved with different numerical methods and numerical grids. Another focus was the treatment of problems posed on complex domains, e.g. on manifold surfaces in space and/or on domains with complex boundaries. In this case, the goal was to treat the effects of inaccuracies and/or uncertainty in the representation of the domain geometry. Finally, we also establshed several rigorous convergence results for a class of goal-oriented adaptive methods that are designed to drivdriving the error in a specific quantity of interest below a given tolerance.

Along with theoretical development, the project studied the practical implementation of a posteriori error estimates for complex physics, including high performance issues. The project also addressed the question of efficient computation. The availability of accurate error estimates raises the ability to develop efficient adaptive error control algorithms in which various discretization parameters are adjusted based on relative contributions to the overall error in order to achieve a desired accuracy with minimal computational work. In another direct, the project expanded a posteriori error estimates for computed distributions and probabilities arising in computational sensitivity analysis and developed generalized adaptive algorithms that allow for balancing all sources of error and uncertainty affecting the analysis.

The project P.I.s' undertook a significant degree of interdisciplinary interaction during the projects in order to insure that project accomplishments would have impact in science and engineering.

1.2. Detailed descriptions of specific accomplishments

In this section, we describe specific technical accomplishments of the project.

A posteriori error analysis for a transient conjugate heat transfer

We analyzed the accuracy of an operator decomposition finite element method for a transient conjugate heat transfer problem consisting of two materials coupled through a common boundary. We derive accurate a posteriori error estimates that account for the transfer of error between compo-

nents of the operator decomposition method as well as the errors in solving the iterative system. We address a loss of order of convergence that results from the decomposition, and show that the order of convergence is limited by the accuracy of the transferred gradient information. We extend a boundary flux recovery method to transient problems and use it to regain the expected order of accuracy in an efficient manner. In addition, we use the a posteriori error estimates to adaptively compute the recovered boundary flux only within the domain of dependence for a quantity of interest.

A posteriori error estimation and adaptive mesh refinement for a multiscale operator decomposition approach to fluid-solid heat transfer

We analyze a multiscale operator decomposition finite element method for a conjugate heat transfer problem consisting of a fluid and a solid coupled through a common boundary. We derive accurate a posteriori error estimates that account for all sources of error, and in particular the transfer of error between fluid and solid domains. We use these estimates to guide adaptive mesh refinement. In addition, we provide compelling numerical evidence that the order of convergence of the operator decomposition method is limited by the accuracy of the transferred gradient information, and adapt a so-called boundary flux recovery method developed for elliptic problems in order to regain the optimal order of accuracy in an efficient manner. In an appendix, we provide an argument that explains the numerical results provided sufficient smoothness is assumed.

Nonparametric density estimation for randomly perturbed elliptic problems

We study the nonparametric density estimation problem for a quantity of interest computed from solutions of an elliptic partial differential equation with randomly perturbed coefficients and data. We derive an efficient method for computing samples and generating an approximate probability distribution based on Lion's domain decomposition method and the Neumann series. We then derive an a posteriori error estimate for the computed probability distribution reflecting all sources of deterministic and statistical errors. Finally, we develop an adaptive error control algorithm based on the a posteriori estimate. we extend the analysis to include a "modeling error" term that accounts for the effects of the resolution of the statistical description of the random variation and modify the adaptive algorithm to adapt the resolution of the statistical description. We also prove some related convergence results.

A posteriori error analysis for cell-centered finite volume methods for semilinear elliptic problems

We conduct a goal-oriented a posteriori analysis for the error in a quantity of interest computed from a cell-centered finite volume scheme for a semilinear elliptic problem. To carry out the analysis, we use an equivalence between the cell-centered finite volume scheme and a mixed finite element method with special choice of quadrature.

Blockwise adaptivity for time dependent problems based on coarse scale adjoint solutions

We describe and test an adaptive algorithm for evolution problems that employs a sequence of "blocks" consisting of fixed, though non-uniform, space meshes. This approach offers the advantages of adaptive mesh refinement but with reduced overhead costs associated with load balancing, re-meshing, matrix reassembly, and the solution of adjoint problems used to estimate discretization error and the effects of mesh changes. We describe several strategies to determine appropriate block discretizations from coarse scale solution information using adjoint-based a posteriori error estimates and demonstrate the behavior of the algorithms in a set of examples.

Conservative discretization and a posteriori error analysis for a cut cell diffusion problems with complex geometry

We study the solution of a diffusive process in a domain where the diffusion coefficient changes discontinuously across a curved interface. We consider discretizations that use regularly-shaped meshes, so that the interface "cuts" through the cells (elements or volumes) without respecting the regular geometry of the mesh. Consequently, the discontinuity in the diffusion coefficients has a strong impact on the accuracy and convergence of the numerical method. This motivates the

derivation of computational error estimates that yield accurate estimates for specified quantities of interest. For this purpose, we adapt the well-known adjoint based *a posteriori* error analysis technique used for finite element methods. In order to employ this method, we describe a systematic approach to discretizing a cut-cell problem that handles complex geometry in the interface in a natural fashion yet reduces to the well-known Ghost Fluid Method in simple cases. We test the accuracy of the estimates in a series of examples.

A measure-theoretic computational method for inverse sensitivity problems

We consider the inverse sensitivity analysis problem of quantifying the uncertainty of inputs to a deterministic map given specified uncertainty in a linear functional of the output of the map. This is a version of the model calibration or parameter estimation problem for a deterministic map. We assume that the uncertainty in the quantity of interest is represented by a random variable with a given distribution and we use the Law of Total Probability to express the inverse problem for the corresponding probability measure on the input space. Assuming that the map from the input space to the quantity of interest is smooth, we solve the generally ill-posed inverse problem by using the Implicit Function Theorem to derive a method for approximating the set-valued inverse that provides an approximate quotient space representation of the input space. We then derive an efficient computational approach to compute a measure theoretic approximation of the probability measure on the input space imparted by the approximate set-valued inverse that solves the inverse problem. We also treat the situation in which the output of the map is determined implicitly and is difficult and/or expensive to evaluate, e.g requiring the solution of a differential equation, and hence the output of the map is approximated numerically. The main goal is an a posteriori error estimate that can be used to evaluate the accuracy of the computed distribution solving the inverse problem taking into account all sources of statistical and numerical deterministic errors. We present a general analysis for the method and then apply the analysis to the case of a map determined by the solution of an initial value problem.

A posteriori analysis of multirate numerical methods for multiscale ordinary differential equations

We analyze a multirate time integration method for systems of ordinary differential equations that present significantly different scales within the components of the model. We interpret the multirate method as a multiscale operator decomposition method and use this formulation to conduct both an *a priori* error analysis and a hybrid *a priori – a posteriori* error analysis. The hybrid analysis has the form of a computable *a posteriori* leading order expression and a provably-higher order *a priori* expression. Both analyses distinguish the effects of the discretization of each component from the effects of multirate solution. The effects on stability arising from the multirate solution are reflected in perturbations to certain associated adjoint operators.

Convergence theory for goal-oriented adaptive methods

In the first of the convergence theory subprojects of the DTRA project, We developed a new convergence theory for a general class of adaptive approximation algorithms for nonlinear operator equations, and then used the theory to obtain convergence, contraction, and optimality results for practical adaptive finite element methods (AFEM) applied to several classes of nonlinear elliptic equations and systems of elliptic equations. The results can be viewed as extending the recent convergence results for linear problems of Morin, Siebert and Veeser, and of Nochetto et. al to more general nonlinear problems (with G. Tsogtgerel and Y. Zhu). We also develop new mathematical results for hierarchical error indicators to drive AFEM algorithms, and establish condition number estimates for appropriate preconditioners (with J. Ovall and R. Szypowski). We have further extended these results to the class of adaptive methods that were the target of this DTRA research probejct: goal-oriented adaptive methods that are designed to drive the error in a quantity of interest below a given tolerance. In 2009, Mommer and Stevenson developed a goal-oriented adaptive method for the Poisson equation, together with rigorous convergence and complexity results for their method, establishing what was apparently the first convergence result for a goal-oriented adaptive method. We have now extended the results of Mommer and Stevenson to goal-oriented

adaptive methods for general linear convection-diffusion elliptic problems (with S. Pollock). In a second manuscript, these results were further extended to a large class of scalar nonlinear problems (with S. Pollock and Y. Zhu). All three articles have now been posted on arXiv, submitted for publication, and are currently in review. All of the techniques are demonstrated for practical problems of interest using the FETK software (see below).

Analysis of multiphysics problems with complex domains

We analyzed a large class of regularized Navier-Stokes and Magnetohydrodynamics (MHD) models in three-dimensional spatial domains, a class which includes the Navier-Stokes equations, the Navier-Stokes-alpha model, the Leray-alpha model, the Modified Leray-alpha model, the Simplified Bardina model, the Navier-Stokes-Voight model, the Navier-Stokes-alpha-like models, and certain MHD models, in addition to representing a larger 3-parameter family of models not previously analyzed. We recovered a number of known results for established models, but also obtained new results for all models in this general family, including existence, regularity, uniqueness, stability, attractor existence and dimension, and existence of determining operators. (J. Nonlinear Science 2009, with E. Lunasin and G. Tsogtgerel.)

We then develop and analyze numerical methods for approximation of stationary and evolution problems on surfaces, including coupled elliptic-parabolic systems. A major theoretical breaktrough was showing how the recent finite element error estimates of Demlow and Dziuk can be recovered from a more general approach involving the analysis of variational crimes in Hilbert complexes, generalizing their results for surface finite elements to arbitrary spatial dimension and to applications involving higher-dimensional differential forms and both linear and nonlinear equations. This generalization was made possible through the use and extension of *finite element exterior calculus (FEEC)*. (Found. Comput. Math. 2012, with A. Stern.) We have now extended this work in FEEC in the direction of time-dependent problems; we completed and submitted a new manuscript in 2012 that extends these results on surface finite element methods to scalar parabolic and hyperbolic problems, including again nonlinear problems (with A. Gillette). We also give an analysis of the singularities in a fundamentally important model in biochemistry, and develop a number of AFEM-based numerical techniques for treating these degenerate features in a provably high-fidelity way (Comm. Comput. Phys. 2012, with J. McCammon, Y. Zhou, Y. Zhu, Z. Yu).

In addition, we have developed and implemented goal-oriented, adjoint-based, a posteriori error estimates for elliptic problems on smooth manifolds. In particular, the estimates take into account the effects of domain curvature on accuracy. We also considered the problem of small random perturbations to the manifold, pointing the way to treat problems in which the domain is determined experimentally or by measurement. This work is nearing completion and will be submitted in Summer 2012 (with W. Newton)

Analysis of elliptic problems on domains with randomly perturbed boundaries

We developed a systematic approach to solve elliptic problems on domains that have randomly perturbed boundaries, after first classifying such problems into several different classes. The results are particularly relevant to situations in which the boundaries are obtained through measurement or are subject to error. The approach avoids the need to remesh each new domain in a random sampling Monte Carlo solution. Moreover, we derive a posteriori error estimates that indicate how random perturbations in the boundary affect the accuracy of computed solutions.

A posteriori error analysis of explicit, IMEX, and truncated Picard iteration time integration methods

Explicit, Implicit/Explicit (IMEX), and truncated Picard iteration time integration methods are widely employed to solve multiphysics applications in defense and department of energy enterprises, e.g. such as reacting flows. Such methods requires significant alterations for a posteriori error analysis in order to describe the effects of these approaches on both stability and accuracy. Therefore, last year we undertook the systematic study of a posteriori error analysis for explicit, truncated Picard iteration, and implicit/explicit (IMEX) time integration methods. For explicit

methods, we introduce special projection operators into the standard finite element formulation for evolution problems. These projection operators are (1) a truncated Taylor expansion computed at a past time node and (2) extrapolation from a interpolatory polynomial using values at a collection of previous nodes. We then alter the a posteriori error analysis to include terms that measure the effects of these projections, yielding distinct "explicit" time integration terms in the a posteriori error analysis. We recently have extended this approach to treat IMEX methods. To analyze truncated Picard iteration methods, we exploit an old result of H. Keller and J. Keller for the "matricant", which is the exponential form of the solution operator of a linear ***non-autonomous*** evolution problem. This provides a way to define the adjoint for a solution obtained by truncated Picard iteration, which we then use in the a posteriori error analysis. We have also extended this analysis to implicit methods that employ Jacobi iteration to solve the systems at each step.

Coupled parabolic-elliptic systems

Estep and Holst collaborated on the development methods and a posteriori error analysis for coupled parabolic-elliptic systems of equations. The main application is on modeling of black holes. A new development in the Holst group has been the extension of their recent work on finite element exterior calculus to parabolic and hyperbolic problems (completed and submitted in 2012), which will provide a very strong mathematical framework for the development of methods and a posteriori analysis for coupled parabolic-elliptic problems. This extension to FEEC is now being combined with our recent work on goal-oriented adaptive methods using a variational framework, by which the elliptic component of the system is combined with implicit time-stepping schemes to provide "constraints" in a Lagrange multiplier formulation. We are able to show convergence for the adaptive scheme, generalizing our recent work on convergence theory for goal-oriented adaptive methods (with S. Pollock, Y. Zhu).

Coupled ordinary differential equation - parabolic differential equation

Estep and Hameed (along with collaborators) derived and implemented a posteriori error estimates for systems of evolution equations consisting of a reaction-diffusion problem posed on a global domain coupled to systems of ordinary differential equations in a collection of small cells partitioning the global domain. The local cell problems model chemical reactions that determine the local physical conditions driving the parabolic problem. The analysis takes into account the iteration error in solving the coupled systems.

New approaches to adaptive error control for evolution problems

Estep and Hameed (along with collaborators) developed new adaptive error control algorithms that take into account cancellation of errors to improve efficiency. The approach identifies periods of time over which there is significant cancellation. Inside the regions, uniform refinements are used to preserve the favorable cancellation, while the time step sizes in the various regions are adjusted according to the contribution to the overall error from the regions.

Implementation of theoretical results

The last major goal in this project is implementation of the theoretical results into the FETK code. For this purpose, we recruited a full time postdoc, Ryan Szypowski, working at UCSD under the supervision of co-PI Michael Holst with responsibility to carry out the implementation and testing. He is being jointly supervised by the PI D. Estep. This FETK deveopment has focused on providing a robust, theory-based convergent adaptive finite element implementation for nonlinear problems which retains linear complexity. This has included work on the following specific components, which have been implemented in both the MATLAB subset FETKLab of the 2D code in FETK as well as in the full 2D/3D code in FETK:

1. The element marking strategy was updated to be based on "Dorfler Marking". Special care was taken to use a linear-time complexity binning approach as opposed to an actual sort. Only this type of marking strategy, which is not often used in practice due to its potential costs unless carefully implemented, allows for establishing both convergence and linear overall computational complexity of the adaptive algorithms.

2. A number of new error estimators were added. They include:

   (a) A hierarchical error estimator based on face-bump functions which was proven in our recent publications to be efficient, reliable, and robust. This work included the addition of a new cubic bump finite element space, which led to a better understanding of how we can improve the finite element space implementation to allow for future additions.

   (b) An error estimator based on the solution of a dual problem, which we refer to as dual-weighted residual (DWR). This implementation involved leveraging the work on the bump-function library above, as well as the development of high-order quadrature rules, and the ability to maintain two distinct and unrelated adaptive meshes during a computation, with quantities being projected back and forth between the meshes as needed.

   (c) An error estimator based on smoothed gradients. This is based on recent work of R. Bank and J. Xu, collaborators of the PIs.

3. W. Newton, co-advised by Estep and Holst, implemented the a posteriori error estimates that account for error in the description of the manifold on which the problem is posed developed in his thesis.

4. A driver application for solving nonlinear problems using inexact Newton solvers based on a multilevel approach was written. This has been used for most of the problems described above.

5. Prior to 2013, FETK and the FETKLab MATLAB subset of FETK were primarily based on linear finite element discretizations, with enough partial support for higher-order elements to allow for the use of e.g. bump functions in error indicators and formulation of dual problems. A general element class was developed in early 2013 to allow for use of any type of Lagrange-type element for either the primarl or dual problem. Both linear and quadratic elements were then implemented and are provided with the FETK code base as element examples. Our recent manuscripts with new convergence results for goal-oriented methods contain a large collection of numerical examples that now exploit this infrastructure to carefully compare a number of adaptive methods based on goal functions (with S. Pollock and Y. Zhu).

2. Training and Professional Development

The support of this project has partially contributed to the training and professional development for three graduate students and three postdocs. This includes specialized research-level instruction and individual mentoring as well as participation in large research group activities directed by the PIs. Students and postdocs were encouraged to participate in professional meetings and to interact with researchers in other universities and in national DOE laboratories as appropriate. Students and postdocs were trained to write and prepare and deliver professional presentations.

Details for the trainees:

- Will Newton received his Ph.D. from CSU in 2011, and then was hired as a Research Scientist Class I in PI Estep's group. His primary focus is a project on multiscale models of new nuclear fuels supported by a contract from Idaho National Laboratory. He has continued to work on research related to this project following up on the work in his thesis. Thesis is "A Posteriori Error Estimates for the Poisson Problem on Closed, Two-Dimensional Surfaces", available from Colorado State University Library.

- Nate Burch received his Ph.D. from CSU in 2011, and then took a two year postdoc position at SAMSI (Statistical and Mathematical Sciences Institute) as part of the Program on Uncertainty Quantification. Thesis is "Probabilistic Foundation of Nonlocal Diffusion

and Formulation and Analysis for Elliptic Problems on Uncertain Domains", available from Colorado State University Library.

- The CSU postdoc Jehanzeb Hameed is in the second year of his position in PI Estep's group. His primary focus is a project on a Department of Energy Uncertainty Quantification project that is jointly conducted with Sandia National Laboratory. Part of his research is related to the activities supported in this project.

- Jonny Serencsa received his Ph.D. from UCSD in 2012, and has been doing pre- and post-doctoral work at UC Davis. His doctoral work was jointly supervised by PI Holst and S. Shkoller at UC Davis, and he is currently working for a startup company in the Bay Area.

- Ryan Szypowski received his Ph.D. from UCSD in 2008, and remained at UCSD working with Holst as a postdoc and then research scientist until 2012. He moved to a tenure-track position in the Mathematics Department at Cal Poly Pomona in Fall 2012.

- Andrew Gillette received his Ph.D. from UT Austin in 2011, and joined Holst's group at UCSD as a postdoctoral fellow in Fall 2011. He helped push forward both the the project involving Ryan Szypowski, and the development of an FEEC-based error analysis framework for parabolic and hyperbolic problems. In Fall 2013, Andrew is starting a tenure-track faculty position in the mathematics department at the University of Arizona.

- Sara Pollock received her Ph.D. from UCSD 2012, and remained at UCSD working with Holst as a postdoc during the 2012-2012 academic year. In Fall 2013, Sara is starting a 3-year named postdoctoral position in the mathematics department at Texas A&M.

3. Dissemination

We have disseminated the research in this project through submission of peer-reviewed research articles, presenting many invited talks at universities and conferences, and publishing software developed in this project for public access. A summary of this activity during this project:

- 53 research articles related to the project research have appeared or are accepted

- 19 research articles related to the project research are currently under review

- 5 book and/or book chapters have appeared or are being written

- 60 invited lectures at universities and professional meetings

Applications to multiscale/multiphysics physical and engineering systems

In conjunction with collaborators in engineering, chemistry and biophysics, we have applied many of the algorithms and techniques for multiphysics and multiscale problems developed in this DTRA-supported research program. Our focus continues to be on applications in material, chemical and biological physics of relevance to DOD, DTRA, and DOE missions. In addition to our publications placed in the mathematics literature, we have placed joint publications from these research collaborations with physical scientists and engineers in a broad spectrum of leading scientific journals to maximize the impact of our results, including: *Physical Review Letters*, *Physical Review D*, *Journal of Nonlinear Science*, *Classical and Quantum Gravity*, *Journal of Chemical Theory and Computation*, *Journal of Cell Science*, *Journal of Structural Biology*, *Biophysical Journal*, *PLoS Computational Biology*, *IMA Journal on Applied Mathematics*, *Computer Aided Geometric Design*, *BIT*, *Applied Numerical Mathematics*, *IEEE Journal on Engineering in Medicine and Biology*, *IEEE Transactions on Biomedical Computing*, *Frontiers in Computational Physiology and Medicine*, *Investigative Ophthalmology and Visual Science*, *Journal of Scientific Computing*, *Journal of Applied Mathematics and Computation*, *Communications in Computational Physics*,

Journal of Molecular Graphics and Modeling, Journal of Physical Chemistry B, Journal of Chemical Physics, Communications in Mathematical Physics, Annals of Nuclear Engineering, Journal of Computational Physics, Acta Biomaterialia, Computer Methods in Applied Mechanics and Engineering, Journal of Engineering Mathematics, and *Foundations of Computational Mathematics.*

4. Products

4.1. Publications, conference papers, and presentations

The following papers were accepted or appeared during March 27, 2009 - September 1, 2009

- *A posteriori analysis and adaptive error control for multiscale operator decomposition methods for coupled elliptic systems I: One way coupled systems*, V. Carey, D. Estep, and S. Tavener, SIAM Journal on Numerical Analysis 47 (2009), 740-761

- *A posteriori error analysis for a transient conjugate heat transfer problem*, D. Estep, S. Tavener, T. Wildey, Finite Elements in Analysis and Design 45 (2009), 263-271

- *Nonparametric density estimation for randomly perturbed elliptic problems I: Computational methods, a posteriori analysis, and adaptive error control*, D. Estep, A. Malqvist, and S. Tavener, SIAM Journal on Scientific Computing 31 (2009), 2935-2959

- *Solving the Einstein constraints on multi-block triangulations using finite elements*, O. Korobkin, B. Aksoylu, M. Holst, E. Pazos, and M. Tiglio, Class. Quant. Grav. 26 (2009), No. 14, 145007 (28 pp). (arXiv:gr-qc/0801.1823)

- *An adaptive finite element method for solving the exact Kohn-Sham equation of density functional theory*, E. Bylaska, M. Holst, and J. Weare, Journal of Chemical Theory and Computation, 5 (2009), pp. 937–948.

- *Finite Element Analysis of Drug Electrostatic Diffusion: Inhibition Rate Studies in N1 Neuraminidase*, Y. Cheng, M. Holst, and J.A. McCammon, Biocomputing 2009: Proceedings of the Pacific Symposium, R.B. Altman, A.K. Dunker, L. Hunter, T. Murray, and T.E. Klein, eds., 2009, pp. 281-292.

- *Three-dimensional reconstruction reveals new details of membrane systems for calcium signaling in the heart*, T. Hayashi, M.E. Martone, Z. Yu, A. Thor, M. Doi, M. Holst, M.H. Ellisman, and M. Hoshijima, J. Cell Sci., Vol. 122 (April, 2009), No. 7, pp. 1005-1013.

- *Rough Solutions of the Einstein Constraints on closed manifolds without near-CMC conditions*, M. Holst, G. Nagy, and G. Tsogtgerel, Comm. Math. Phys., Vol. 288 (June 2009), No. 2, pp. 547-613. (arXiv:gr-qc/0712.0798)

- *Multi-Scale Modeling of Ventricular Myocytes: Contributions of structural and functional heterogeneities to excitation-contraction coupling in the normal and failing rodent heart*, S. Lu, A. Michailova, J. Saucerman, Y. Cheng Z. Yu, T. Kaiser, W. Li, R. Bank, M. Holst, A. McCammon, T. Hayashi, M. Hoshijima, P. Arzberger, and A. McCulloch, IEEE Journal on Engineering in Medicine and Biology, Vol. 28 (March-April 2009), No. 2, pp. 46-57.

- *Convergence and Optimality of Adaptive Mixed Finite Element Methods*, L. Chen, M. Holst, and J. Xu, Math. Comp., Vol. 78 (2009), No. 265, pp. 33-53.

The following papers were accepted or appeared during September 2, 2009 - September 1, 2010

- *Nonparametric density estimation for randomly perturbed elliptic problems II: Applications and adaptive modeling*, D. Estep, A. Malqvist, S. Tavener, International Journal for Numerical Methods in Engineering 80 (2009), 846-867

- *A posteriori error analysis of a cell-centered finite volume method for semilinear elliptic problems*, D. Estep, M. Pernice, D. Pham, S. Tavener, H. Wang, Journal of Computational and Applied Mathematics 233 (2009), 459 - 472

- *A posteriori error estimation and adaptive mesh refinement for a multi-discretization operator decomposition approach to fluid-solid heat transfer*, D. Estep, S. Tavener, T. Wildey, Journal of Computational Physics 229 (2010), 4143 - 4158

- *Blockwise adaptivity for time dependent problems based on coarse scale adjoint solutions*, V. Carey, D. Estep, A. Johansson, M. Larson, and S. Tavener, SIAM Journal on Scientific Computing 32 (2010), 2121 - 2145

- *Numerical analysis of Ca2+ signaling in rat ventricular myocytes with realistic transverse-axial tubular geometry and inhibited sarcoplasmic reticulum*, Y. Cheng, Z. Yu, M. Hoshijima, M. Holst, A. McCulloch, and J. M. ad A.P. Michailova, PLoS Computational Biology, 6 (2010), pp. e1000972:1–16.

- *Poisson-Nernst-Planck equations for simulation biomolecular diffusion-reaction processes I: Finite element solutions*, B. Lu, M. Holst, J. McCammon, and Y. Zhou, J. of Comput. Phys. 229 (2010), 6679-7794 (16 pp).

- *Analysis of a general family of regularized Navier-Stokes and MHD models*, M. Holst, E. Lunasin, and G. Tsogtgerel, J. Nonlin. Sci., 20 (2010), pp. 523–567.

The following book chapter appeared during September 2, 2009 - September 1, 2010

- *Error estimation for multiscale operator decomposition for multiphysics problems*, D. Estep, Chapter 11, in *Bridging the Scales in Science and Engineering*, J. Fish, editor, Oxford University Press, 2010

The following books were under contract or appeared during September 2, 2009 - April 5, 2013

- *Practical Analysis in Many Variables*, D. Estep, SIAM, 2010.

- *Green's Functions and Boundary Value Problems, Third Edition*, I. Stakgold and M. Holst, John-Wiley, 888 pages, February 2011.

The following nonrefereed papers appeared during September 2, 2009 - September 1, 2010

- *CSE 2009: Graduate Education in CSE - Structure for the Zoo?*, H.-J. Bungartz and D. Estep, SIAM News 42, 2009

- *Computational Science and Engineering Education: SIAM's Perspective*, H.-J. Bungartz, D. Estep, U. Rude, and P. Turner, IEEE Computing in Science and Engineering 11 (2009), 5-11

- *Interview with Chief Editor of the SIAM CSE Book Series*, D. Estep, SIAM News 43 (2010)

The following papers were accepted or appeared during September 2, 2010 - September 1, 2011

- *A computational measure theoretic method for inverse sensitivity problems I: Basic method and analysis*, J. Breidt, T. Butler, and D. Estep, SIAM Journal on Numerical Analysis, 2011, 49 (2011), 1836-1859
- *A posteriori error analysis for a cut cell finite volume method*, D. Estep, S. Tavener, M. Pernice, H. Wang, Computer Methods in Applied Mechanics and Engineering, 2010, 233 (2009), 459-472
- *Parameter estimation and directional leverage with applications in differential equations*, N. Burch, D. Estep, and J. Hoeting, Metrica, Metrika, DOI: 10.1007/s00184-011-0358-4, 2011
- *Continuum Modeling and Control of Large Mobile Networks*, Y. Zhang, E. K. P. Chong, J. Hannig, and D. Estep, Proceedings of the 49th Annual Allerton Conference on Communication, Control and Computing, Illinois, 2011
- *Nonparameteric density estimation for randomly perturbed elliptic problems III: Convergence, complexity, and generalizations*, D. Estep, M. Holst, and A. Malqvist, Journal of Applied Mathematics and Computing 38 (2012), 367-387
- *An efficient, reliable and robust error estimator for elliptic problems in* $\mathbb{R}^3$, M. Holst, J. Ovall, and R. Szypowski, Applied Numerical Mathematics, 61 (2011), 675695
- *Efficient mesh optimization schemes based on optimal delaunay triangulations*, L. Chen and M. Holst, Computer Methods in Applied Mechanics and Engineering 200 (2011), 967984
- *Adaptive finite element modeling techniques for the Poisson-Boltzmann equation*, M. Holst, J. McCammon, Z. Yu, Y. Zhou, and Y. Zhu, Communications in Computational Physics, 11 (2012), pp. 179–214.
- *Convergence analysis of finite element approximations of the Joule heating problem in three spatial dimensions*, M. Holst, M. Larson, A. Malqvist, and R. Soderlund, BIT, 50 (2010), pp. 781–795.
- *Semilinear mixed problems on Hilbert complexes and their numerical approximation*, M. Holst AND A. Stern, Foundations of Computational Mathematics, 2010, 12 (2012), pp. 363–387
- *Adaptive solution of the Poisson-Boltzmann equation using goal-oriented error indicators*, B. Aksoylu, S. Bond, E. Cyr, AND M. Holst, J. Sci. Comput. 52 (2012), 202-225 (23 pp).

The following papers were accepted or appeared during September 2, 2011 - September 1, 2012

- *A computational measure theoretic approach to inverse sensitivity problems II: A posteriori error analysis*, T. Butler, D. Estep and J. Sandelin, SIAM Journal on Numerical Analysis, 50 (2012)
- *Viscoelastic Effects During Loading Play an Integral Role in Soft Tissue Mechanics*, K. Troyer, D. Estep, and C. Puttlitz, Acta Biomaterialia 8 (2012), 234-244
- *A posteriori analysis of multirate numerical method for ordinary differential equations*, D. Estep, V. Ginting, S. Tavener, 2012, Computer Methods in Applied Mechanics and Engineering, 223-224 (2012), 10-27
- *Adaptive error control for an elliptic optimization problem*, Applicable Analysis, D. Estep and S. Lee, 2012, DOI:10.1080/00036811.2012.683785, 1-15

- *Analysis of routing protocols and interference-limited communication in large networks via continuum modeling*, N. Burch, E. Chong, D. Estep, J. Hannig, Journal of Engineering Mathematics, 2012, (DOI) 10.1007/s10665-012-9566-9

- *A numerical method for solving a stochastic inverse problem for parameters*, T. Butler and D. Estep, Annals of Nuclear Energy, 2012, 10.1016/j.anucene.2012.05.016

- *Geometric variational crimes: Hilbert complexes, finite element exterior calculus, and problems on hypersurfaces*, M. Holst and A. Stern, Foundations of Computational Mathematics, 12 (2012), pp. 263–293.

- *Multi-scale modeling of calcium dynamics in ventricular myocytes with realistic transverse tubules*, Z. Yu, G. Yao, M. Hoshijima, A. Michailova, and M. Holst, IEEE TBME Letters, Special Issue on Multi-Scale Modeling and Analysis for Computational Biology and Medicine, 58 (2011), No. 10, 2947-2951 (4 pp).

- *Multiscale continuum modeling and simulation of biological processes: From molecular electro-diffusion to sub-cellular signaling transduction*, Y. Cheng, M. Holst, J. McCammon, and A. Michailova, Comput. Sci. Disc., 5 (2012), 015002-015015 (13 pp).

- *The Navier-Stokes-Voight model for image inpainting*, M. Ebrahimi, M. Holst, and E. Lunasin, IMA J. Appl. Math., doi:10.1093/imamat/hxr069 (2012), 1-26 (26 pp).

- *Numerical bifurcation analysis of conformal formulations of the Einstein constraints*, M. Holst and V. Kungurtsev, Phys. Rev. D, 84 (2011), pp. 124038(1)–124038(8).

- *Modeling cardiac calcium sparks in a three-dimensional reconstruction of a calcium release unit*, J. Hake, A. Edwards, Z. Yu, P. Kekenes-Huskey, A. Michailova, A. McCammon, M. Holst, M. Hoshijima, and A. McCulloch, J. Physiol., 590 (2012), No. 18, 4403-4422 (18 pp).

- *Localized glaucomatous change detection within the proper orthogonal decomposition framework*, M. Balasubramanian, D. Kriegman, C. Bowd, M. Holst, R. Winreb, P. Sample, and L. Zangwill, Invest. Ophthalmol. Vis. Sci., 53 (2012), No. 7, 3615-3628 (14 pp).

- *Quality tetrahedral mesh smoothing via boundary-optimized Delaunay triangulation*, Z. Gao, Z. Yu, and M. Holst, Computer Aided Geometric Design, 29(9):707-721, 2012.

- *Modeling effects of L-type Ca2+ current and Na+-Ca2+ exchanger on Ca2+ trigger flux in rabbit myocytes with realistic T-tubule geometries*, P. Kekenes-Huskey, Y. Cheng, J. Hake, F. Sachse, J. Bridge, M. Holst, J. McCammon, A. McCulloch, and A. Michailova, Frontiers in Physiology, 3 (2012), pp. 1–14.

The following papers were accepted, appeared or were submitted and still pending review during September 2, 2011 - September 1, 2012

- *A Posteriori Analysis and Adaptive Error Control for Multiscale Operator Decomposition Solution of Elliptic Systems II: Fully Coupled Systems*, V. Carey, D. Estep, S. Tavener, International Journal of Numerical Methods in Engineering, 2011, in revision

- *A posteriori analysis of an iterative multi-discretization method for reaction-diffusion systems*, J. H. Chaudhry, D. Estep, V. Ginting, and S. Tavener, Computer Methods in Applied Mechanics and Engineering, 2012, in revision

- *A-posteriori error estimates for mixed finite element and finite volume methods for problems coupled through a boundary with non-matching grids*, T. Arbogast, D. Estep, B. Sheehan, and S. Tavener, IMA J. Numerical Analysis, 2012, in revision

- *Multilevel preconditioners for discontinuous Galerkin approximations of elliptic problems with jump coefficients*, B. Ayuso de Dios, M. Holst, Y. Zhu, and L. Zikatanov, in Proceedings of the Twentieth International Conference on Domain Decomposition Methods, San Diego, USA, San Diego, CA, USA, February 2011.

- *Local multilevel preconditioners for elliptic equations with jump coefficients on bisection grids*, L. Chen, M. Holst, J. Xu, and Y. Zhu, Submitted for publication.

- *Local convergence of adaptive methods for nonlinear partial differential equations*, M. Holst, G. Tsogtgerel, and Y. Zhu, Submitted for publication.

- *The Lichnerowicz equation on compact manifolds with boundary*, M. Holst and G. Tsogtgerel, Submitted for publication.

- *Adaptive finite element methods with inexact solvers for the nonlinear Poisson-Boltzmann equation*, M. Holst, R. Szypowski, and Y. Zhu, in Proceedings of the Twentieth International Conference on Domain Decomposition Methods, San Diego, USA, San Diego, CA, USA, February 2011.

- *Barrier methods for critical exponent problems in geometric analysis and mathematical physics*, J. Erway and M. Holst, Submitted for publication.

- *Finite element error estimates for critical exponent semilinear problems without angle conditions*, R. Bank, M. Holst, R. Szypowski, and Y. Zhu, Submitted for publication.

- *Convergence and optimality of goal-orientied adaptive finite element methods for nonsymmetric problems*, M. Holst and S. Pollock, Submitted for publication.

- *Generalized solutions to semilinear elliptic PDE with applications to the Lichnerowicz equation*, M. Holst and C. Meier, Submitted for publication.

- *Finite element exterior calculus for evolution problems*, A. Gillette and M. Holst, Submitted for publication.

- *Two-grid methods for semilinear interface problems*, M. Holst, R. Szypowski, and Y. Zhu, Accepted for publication in Numer. Methods Partial Differtial Equations.

- *Convergence of goal-oriented adaptive finite element methods for semilinear problems*, M. Holst, S. Pollock, and Y. Zhu, Submitted for publication.

- *Feature-preserving surface mesh smoothing via suboptional Delaunay triangulation*, Z. Gao, Z. Yu, and M. Holst, Graphical Models, 75 (2013), pp. 23–38.

The following papers were accepted, appeared or were submitted and still pending review during September 2, 2012 - April 5, 2012

- *Multiphysics Simulations: Challenges and Opportunities,* D. E. Keyes, L. C. McInnes, C. Woodward, W. Gropp, E. Myra, M. Pernice, J. Bell, J. Brown, A. Clo, J. Connors, E. Constantinescu, D. Estep, K. Evans, C. Farhat, A. Hakim, G. Hammond, G. Hansen, J. Hill, T. Isaac, X. Jiao, K. Jordan, D. Kaushik, E. Kaxiras, A. Koniges, K. Lee, A. Lott, Q. Lu, J. Magerlein, R. Maxwell, M. McCourt, M. Mehl, R. Pawlowski, A. Peters Randles, D. Reynolds, B. Riviere, U. Ruede, T. Scheibe, J. Shadid, B. Sheehan, M. Shephard, A. Siegel, B. Smith, X. Tang, C. Wilson, and B. Wohlmuth, International Journal of High Performance Computing Applications (27), 2013.

- *Continuum Modeling and Control of Large Nonuniform Wireless Networks via Nonlinear Partial Differential Equations,* Y. Zhang, E. Chong, J. Hannig, and D. Estep, Abstract and Applied Analysis (16), 2013, doi:10.1155/2013/262581, 1-16
- *A posteriori error estimates for explicit time integration methods,* J. Collins, D. Estep and S. Tavener, BIT Numerical Mathematics, 2012, submitted
- *Continuum Limits of Markov Chains with Application to Wireless Network Modeling,* Y. Zhang, E. Chong, J. Hannig, and D. Estep, IEEE Access, 2013, submitted
- *A posteriori error estimation for the Lax-Wendroff finite difference scheme,* J. B. Collins, D. Estep, and S. Tavener, Journal of Computational and Applied Mathematics, 2013, submitted
- *Convergence and optimality of adaptive methods in the Finite Element Exterior Calculus framework*, M. Holst, A. Mihalik, and R. Szypowski, Submitted for publication.
- *An alternative between non-unique and negative yamabe solutions to the conformal formulation of the einstein constraint equations*, M. Holst and C. Meier, Submitted for publication.
- *Non-uniqueness of solutions to the conformal formulation*, M. Holst and C. Meier, Submitted for publication.
- *Efficient computational in multiscale geometric modeling for biomolecular complexes*, T. Liao, Y. Zhang, P. Kekenes-Huskey, A. Michailova, M. Holst, and J. A. McCammon, Submitted for publication.
- *Multilevel preconditioners for discontinuous Galerkin approximations of elliptic problems with jump coefficients*, B. Ayuso de Dios, M. Holst, Y. Zhu, and L. Zikatanov, Accepted for publication in Math. Comp.

4.2. Presentations at meetings, conferences, seminars

The following presentations were made during March 27, 2009 - September 1, 2009

Burch: Research Seminar, Sandia National Laboratory, Albuquerque, New Mexico, 8/09

Estep: Computational Science and Engineering (CSE) Annual Research Symposium, University of Illinois, Urbana-Champaign, Keynote Speaker, 4/09

Estep: SIAM Annual Meeting, Minisymposium on Predictive Computational of Multiscale-Multiphysics Applications, invited speaker, 7/09

Estep: Workshop on Simulating the Spatial-Temporal Patterns of Anthropogenic Climate Change, Los Alamos Institute for Advanced Studies, Santa Fe, New Mexico, invited speaker, 8/09

Estep: Colloquium, Department of Mathematics, University of Wyoming, 9/09

Holst 25th Pacific Coast Gravity Meeting (PCGM25), Eugene, Oregon, 4/09

Holst: 5th Annual Structured Integrators Workshop, Caltech, Pasadena, California, Plenary Speaker, 5/09

Holst: FEniCS 2009 Workshop, Oslo, Norway, Plenary Speaker, 6/09

Holst: Numerische Mathematik 50, Munich, Germany, Plenary Speaker, 6/09

Holst: Mathematical and Numerical Geometric Analysis Workshop, Frieburg, Germany, Plenary Speaker, 9/09

Holst: ICNAAM Conference, Crete, Greece, Minisymposium Speaker, 9/09

Serencsa: CSME Seminar Series, UC San Diego, San Diego, California, 6/09

The following presentations were made during September 2, 2009 - September 1, 2010

Burch: *ICMS Workshop on Uncertainty Quantification*, Edinburgh, UK, 05/10

Estep: *Workshop on Adaptive and Multilevel Methods for Partial Differential Equations*, University of California San Diego, 11/09

Estep: Seminar, Lawrence Livermore National Laboratory, 12/09

Estep: Colloquium, Department of Atmospheric Science, Colorado State University, 1/10

Estep: Seminar, University of Wisconsin, 2/10

Estep: Seminar, Brown University, 3/10

Estep: Seminar, University of Chicago, 3/10

Serencsa: CCoM Seminar Series, UC San Diego, San Diego, California, 11/09

Holst: Plenary Lecture, Symposium on Mathematical Systems Biology, UCI, Irvine, California, 1/10

Holst: Lecture, 26th Pacific Coast Gravity Meeting (PCGM26), San Diego, CA, 3/10

Holst: Plenary Lecture, Workshop on Unstructured Mesh Methods in Mathematical Physics, Jena, Germany, 8/10

Holst: Invited Lecture, Department of Mathematics, Free University of Berlin, Berlin, Germany, 8/10

Holst: Invited Lecture, Department of Mathematics, Technical University of Berlin, Berlin, Germany, 8/10

Holst: Invited Lecture, Department of Mathematics, Jacobs University, Brehmen, Germany, 9/10

The following presentations were made during September 2, 2010 - September 1, 2011

Estep: *SIAM Computational Science and Engineering Conference*, Minisymposia on Numerical Discretization Error Estimation for Uncertainty Quantification, Progress in Computational Methods and Software for Tightly-coupled Multiphysics Applications, Numerical Methods for Stochastic Computation and Uncertainty Quantification, Numerical Challenges in Microstructure Modeling for Materials Science, Reno, Nevada, 2011

Estep: Seminar, Lawrence Livermore National Laboratory, 9/10

Estep: Seminar, Purdue University, 9/10

Estep: Seminar, North Carolina State University, 11/10

Estep: Seminar, Lawrence Livermore National Laboratory, 1/11

Estep: Seminar, University of Southern California, 3/11

Estep: Plenary Talk, ICiS Workshop on Multiphysics Simulations: Challenges and Opportunities, Park City, Utah, 8/11

Holst: Invited Lecture, Department of Mathematics, Jacobs University, Bremen, Germany, 9/10

Holst: Invited Lecture, Workshop on Latest Trends and Developments in Computational Technology and Methods for Solids, Structures, Fluids and Fluid-Structure Interaction, La Jolla, CA, 9/10

Holst: Invited ICES Lecture, University of Texas, Austin, TX, 2/11

Holst: Invited CVS Lecture, University of Texas, Austin, TX, 2/11

Holst: Colloquium, Department of Mathematics, University of Wisconsin, Madison, WI, 4/11

Holst: Colloquium, Department of Mathematics, The Penn State University, State College, PA, 4/11

Holst: Colloquium, Department of Applied Mathematics, University of Washington, Seattle, WA, 5/11

Holst: Seminar, Pacific Northwest National Laboratory, Richland, WA, 5/11

Holst: Plenary Lecture, Workshop on Advances and Challenges in Computational General Relativity, Brown University, Providence, RI, 5/11

Holst: Invited Lecture, Schnelle Löser für partielle Differentialgleichungen, Mathematisches Forschungsinstitut Oberwolfach, Oberwolfach, Germany, 5/11

The following presentations were made during September 2, 2011 - September 1, 2012

Estep: Invited Lecture, Uncertainty Quantification for High-Performance Computing Workshop, Oak Ridge National Laboratory, 5/12

Estep: Invited Lecture, 6th International Conference on Automatic Differentiation, Fort Collins, CO, 7/12

Estep: Invited Paper, Joint Statistical Meetings, 8/12

Estep: Invited Seminar, University of Chicago, 9/11

Estep: Invited Seminar, Florida State University, 4/12

Estep: Invited Seminar, Colorado School of Mines, 4/12

Estep: Invited Colloquium, Statistical and Applied Mathematical Sciences Institute (SAMSI), 4/12

Holst: Invited Lecture, Workshop on Geometric Partial Differential Equations: Theory, Numerics and Appli- cations, Mathematisches Forschungsinstitut Oberwolfach, Oberwolfach, Germany, 11/11

Holst: Invited Lecture, JTO Faculty Fellowship Lecture (1 of 2), Institute for Computational Engineering and Science (ICES), University of Texas, Austin, TX, 11/11

Holst: Invited Lecture, JTO Faculty Fellowship Lecture (2 of 2), Institute for Computational Engineering and Science (ICES), University of Texas, Austin, TX, 1/12

Holst: Plenary Lecture, CSU Research Colloquium, Physics at CSU: Neutrinos to Nano Science, Colorado State University, Fort Collins, CO, 3/12

Holst: Plenary Lecture, 21st International Conference on Domain Decomposition Methods, Rennes, Frances, 6/12

The following presentations were made during September 2, 2012 - April 5, 2013

Pollock: Center for Computational Mathematics Seminar, UCSD, San Diego, CA, 1/13.

Pollock: Joint MAA-AMS Mathematics Meetings, San Diego, CA, 1/13.

Pollock: Numerical analysis seminar, Texas A&M University, College Station, TX, 4/13.

Pollock: CSME Seminar, UCSD, San Diego, CA, 4/13.

Pollock: Minisymposium Lecture, SIAM Annual Meeting, San Diego, CA, 7/13.

4.3. Websites

Research results and software are presented at

- http://www.stat.colostate.edu/~estep/
- http://ccom.ucsd.edu/~mholst/

4.4. Technologies and techniques

Over the last several years, our DTRA-supported research team has led the development of the Finite Element ToolKit, which is an opensource finite element modeling toolkit designed for the simulation of coupled multiphysics problems with multiscale phenomena. The software has been designed and developed collaboratively by both Holst and Estep, and consists of a collection of object-oriented class libraries written in C, C++, Objective C, and Python. There is also a MATLAB/Octave-based prototyping tool (FETKLab), the development of which has been done by both Estep and Holst, as well as several of their graduate students. FETK (and FETKLab) are designed to adaptive discretize and solve coupled reaction-diffusion systems, and is based around state-of-the-art algorithms for simplex mesh generation, error estimation, mesh refinement, finite element discretization, iterative nonlinear and optimization techniques, and fast multilevel and domain decomposition-based linear solvers and preconditions. Many of the algorithms developed in our research articles as described in this report have been prototyped, implemented, and applied to applications in conjunction with physical scientists using FETK. The entire FETK source tree was released in June 2010 on the FETK.org website, as a major milestone of this DTRA project. A substantial extention to both FETK and FETKLab was completed in Spring 2013 that added general Lagrange-type elements for either primal or dual problems, and this new capability has been exploited in a number of our recent articles.

In addition, we continue development on GAASP (Globally Accurate, Adaptive Sensitivity analysis Package) to extend its capabilities for both forward and inverse stochastic sensitivity analysis of differential equations.

5. Impact

5.1. Impact on the principal disciplines of the project

The numerical solution of multiscale, multiphysics models on complex domains along with the development of tools for predictive science and uncertainty quantification is one of the grand challenges facing the mathematical sciences at present. Such problems present a very complex picture in terms of stability and important behaviors interacting across a wide range of scales, which makes the straightforward use of classical numerical methods and analyses extremely problematic, if not impossible. Classic approaches were developed in the context of models involving single physics phenomena operating at a narrow range of scales. While building on classic approaches, the research in this project contributes at a fundamental theoretical level by laying the foundation for reliably accurate and efficient numerical solution based on a posteriori error analysis that accounts for the numerical complexities involved with simulating such systems. This is achieved

by combining extremely sophisticated mathematics in analysis and geometry with cutting edge numerical methodology.

The impact of the research related to this project is widespread, as can be seen in the greatly increasing levels of activity around the world on such problems. This is also evidenced by the number of invitations to speak, the number of funded interdisciplinary projects including a recent award of an extremely prestigious National Science Foundation Focused Research Group (FRG) award to Estep and Holst, the citation record (Estep's h-index is 15 and Holst's h-index is 20), and the high level of the involvement of the PI's in research environment through panels, reports, editing, and so on.

5.2. Impact on other disciplines

Developing reliable and accurate tools for carrying out predictive science and engineering for multiscale, multiphysics systems on complex domains and conducting uncertainty quantification in simulated results is the major problem of computational science and engineering at present. Addressing this challenge requires fundamental research in the mathematical sciences. This project is aimed at addressing a number of key research problems involved with simulating multiphysics systems. Along with theory, the PIs systematically implement the results into public software, and, along with their collaborators, use the software to tackle scientific and engineering research problems. This yields a direct transfer of the theoretical mathematical developments and software implementations to the application domain.

This is evidenced by the large number of interdisciplinary collaborations of the PIs and the substantial volume of interactions with Department of Energy laboratories and industry. Details are provided below.

5.3. Impact in the profession

5.4. Honors and awards

Estep was appointed (founding) Co-Editor in Chief of the SIAM / ASA Journal on Uncertainty Quantification

Estep won the University Scholarship Impact Award, Colorado State University, 2011

Estep was appointed University Interdisciplinary Research Scholar, Colorado State University in 2009

Estep received the Oliver P. Pennock Distinguished Service Award, Colorado State University in 2009

Estep was appointed Editor in Chief, SIAM Book Series on Computational Science and Engineering, 2009 - 2014

Holst received the CSU Distinguished Alumnus Award, 2009

Holst was appointed the Chancellor's Associates Endowed Chair in Mathematics and Physics at UC San Diego in 2012

5.5. Impact on the professional research community

Estep served as one of the Moderators for the SAMSI National SIAM and ASA Town Hall Meeting on Uncertainty Quantification, 2010

Estep served as the Co-Organizer and first Chair, SIAM Activity Group on Uncertainty Quantification, 2010

Estep served as a Program Leader for the SAMSI Program on Uncertainty Quantification, 2011-2012

Estep served as co-chair of the first SIAM/ASA/USACM Conference on Uncertainty Quantification (April, 2012)

Estep along with J. Berger (Duke) and M. Gunzburger (FSU) proposed a new Journal on Uncertainty Quantification to be jointly published by the ASA and SIAM

Estep serves on the Advisory Board for the Center for Advanced Modeling and Simulation, Idaho National Laboratory, 2009 - 2012

Estep serves on the Governing Board of the Statistical and Applied Mathematical Sciences Institute (SAMSI), 2009-2016

Estep served on the National Science Foundation Office of Cyberinfrastructure Grand Challenges Communities Task Force, 2009-2010 (co-author of final recommendation report)

Estep served as Breakout Lead and Report co-author, Uncertainty Quantification and Stochastic Systems, Department of Energy Cross-Cutting Technologies for Computing at the Exascale, 2010

Estep was an invited participant in the Fusion Simulation Program Definition Workshop, 2011

Estep serves on the American Mathematical Society Simmons Travel Grants Committee, 2011-2014

Estep serves as Moderator, Mathematics in the Geosciences Workshop, Northwestern University, 2011

Estep was co-author of *Multiphysics Simulations: Challenges and Opportunities*, Tech. Report ANL/MCS-TM-321, Argonne National Laboratory, 2011

Estep was co-author of *Fostering Interactions Between the Geosciences and Mathematics, Statistics, and Computer Science*, Technical Report TR-2012-02, Department of Computer Science, University of Chicago, 2012

Holst serves on the Executive Committee for the San Diego Supercomputer Center (SDSC), 2007-present

Holst is a Co-Organizer (with R. Bank) of 20th International Conference on Domain Decomposition (DD20), February 2011.

Holst is the Primary Organizer (with J. Hameed): Numerical Methods for Implicit Models in Biomolecular Systems, SIAM CS&E Conference Minisymposium, March 2011

Holst is the Primary Organizer (with A. Demlow, A. Gillette, Y. Zhu): Workshop on Exploiting Geometry in the Development of Numerical Methods for Partial Differential Equations, UCSD Workshop, San Diego, November 2011.

Holst is the Primary Organizer (with A. Demlow, R. Szypowski): Exploiting Geometry in the Development of Numerical Methods for Partial Differential Equations, SIAM Analysis of PDE Conference Minisymposium, November 2011.

Holst is the Primary Organizer (with D. Arnold, A. Gillette): AMS Joint Meeting FEEC Minisymposium, on New Developments in the Finite Element Exterior Calculus, January 2013.

Holst is the Primary Organizer (with A. Gillette, R. Szypowski): Workshop on Exploiting Geometry in the Development of Numerical Methods for Partial Differential Equations II, UCSD Workshop, San Diego, January 2013.

Holst and Estep regularly serve on Grant Review Panels for NSF and DOE, 2004-present

5.6. Professional editorial appointments

Estep: co Editor in Chief (founding), SIAM / ASA Journal on Uncertainty Quantification

Estep: Editor in Chief, SIAM Book Series on Computational Science and Engineering, 2009 - 2014

Estep: Associate Editor, SIAM Journal on Numerical Analysis, 2005-2011

Estep: Associate Editor, International Journal for Uncertainty Quantification, 2010-

Estep: Associate Editor, Multiphysics Modeling Book Series, A. A. Balkema Publishing, CRC Press, 2010-

Estep: Associate Editor, Journal of Applied Mathematics and Computing, 2008-2013

Holst: Associate Editor, Numerische Mathematik, 2008-present

Holst: Associate Editor, SIAM Book Series on Computational Science and Engineering, 2009-2014

5.7. Impact on technology transfer

The PIs maintain a very substantial interdisciplinary collaboration activity with scientists and engineers in universities, Department of Energy laboratories, and industry. These collaborations lead to direct injection of research ideas into practical use.

5.8. Consulting and collaborative activities

In this section, we report currently funded projects that involve substantial interdisciplinary collaborations and transfer of research results related to this project into applications.

Estep is co-PI on the project *Framework Application for Core-Edge Transport Simulations (FACETS)* funded by the Office of Advanced Scientific Computing Research and Office of Fusion Energy Sciences, Department of Energy, 2007-12. Collaborators include: R. H. Cohen, L. Diachin, and T. Epperly at Lawrence Livermore National Laboratory; J. Larson and L. McInnes at Argonne National Laboratory; M. R. Fahey and J. Cobb at Oak Ridge National Laboratory. Subject is development and analysis of numerical solution methods for coupled core-edge fusion simulations.

Estep is PI on the project *Collaborative Proposal: Transforming How Climate System Models are Used: A Global, Multi-Resolution Approach to Regional Ocean Modeling* funded by the Department of Energy, 2009-11. Collaborators include Todd Ringler at Los Alamos National Laboratory. Subject is development and analysis of numerical methods for multiscale ocean models.

Estep is PI on the project *Adjoint-based methods for uncertainty quantification* funded by the Lawrence Livermore National Laboratory, 2010-13. Collaborators are Carol Woodward and Jeff Hittinger at Lawrence Livermore National Laboratory. Duties include (1) pursue develop a posteriori error estimates for hyperbolic problems including shock behavior and (2) consult on uncertainty and error quantification with laboratory personnel

Estep is co-PI on the project *The Inverse Problem for Estimation of Structure of Biological Macromolecules from Small-Angle X-Ray Scattering* funded by the National Institutes of Health, 2010-2014. Collaborators include Jay Breidt (Statistics, CSU) and Karolin Luger (Biochemistry, CSU). Subject is determining the structure of biological macromolecules using small angle x-ray scattering data.

Estep is PI on the project *Enabling Predictive Simulation and UQ of Complex Multiphysics PDE Systems by the Development of Goal-Oriented Variational Sensitivity Analysis and a-Posteriori Error Estimation Methods* funded by the Department of Energy, 2010-2013. Collaborators include John Shadid (Sandia Nat. Lab.) and Victor Ginting (U. Wyom.). Subject is developing a posteriori error estimates for solutions of reacting flow and fusion reaction models.

Estep is co-PI on the project *Collaborative Research: A posteriori error analysis and adaptivity for discontinuous interface problems* funded by the National Science Foundation, 2010-2013. Collaborator is Simon Tavener (CSU). Purpose is developing and analyzing conservative solution methods for elliptic problems with coefficients that are discontinuous across complex interfaces.

Estep is PI on the CSU Subcontract from Multiscale Design Systems, LLC supported by an Air Force SBIR Phase II grant. Collaborators are Simon Tavener (CSU) and Jacob Fish (Columbia Uni.) in 2011. Purpose is developing fast methods for UQ for multiscale models of polymers in stressed environments.

Estep is PI on the project *Uncertainty Analysis for Multiscale Models of Nuclear Fuel Performance* supported by the Idaho National Laboratory from 2011-2014. Collaborators are Simon Tavener (CSU) and Michael Pernice (Idaho Nat. Lab.). Purpose is UQ for multiscale models of nuclear fuel.

Estep is PI on the project *11-2031: Multiscale modeling and uncertainty quantification for nuclear fuel performance*, Nuclear Energy University Programs, Department of Energy, 2011-14. Collaborators are Simon Tavener (CSU), Michael Pernice (INL), Peter Polyakov (Wyoming), Dongbin Xiu (Purdue), Anter el Azab (Purdue)

Estep is a co-PI on the project *Data-Driven Inverse Sensitivity Analysis for Predictive Coastal Ocean Modeling*, Computational and Data-Enabled Science and Engineering in Mathematical and Statistical Sciences (CDS&E-MSS), National Science Foundation, 2012-15. Collaborators are Troy Butler (CSU), Clint Dawson (U. Texas at Austin), and Joannes Westerink (Notre Dame)

Estep and Holst are co-PIs on the project *FRG: Error Quantification and Control for Gravitational Waveform Simulation* funded by the National Science Foundation, 2011-2014. The Project is concerned with estimating the error in computed wave forms obtained from LIGO data.

Holst is PIs on the project *FRG: Analysis of the Einstein Constraint Equations* funded by the National Science Foundation, 2013-2016. The Project is concerned with further extending the solution theory for the Einstein constraint equations.

Holst is PI on the project *MRI: Acquisition of a Parallel Computing and Visualization Facility to Enable Integrated Research and Training in Modern Computational Science, Mathematics, and Engineering* funded by National Science Foundation, 2008-2011. Collaborators include Randolph Bank (UCSD Mathematics), Scott Baden (UCSD Computer Science), and John Weare (UCSD Chemistry). The subject is the design and construction of a state-of-the-art parallel computing system with an excess of 1000 compute nodes, Infiniband high-speed network fabric, parallel filesystems, LCD vizualization walls, housed in a modern server room with raised floor and forced chilled air.

Holst is PI on the project *Adaptive Methods and Finite Element Exterior Calculus for Nonlinear Geometric PDE*, funded by National Science Foundation, 2012-2015. Co-PI is former student and postdoc Ryan Szypowski, now an assistant professor in mathematics at Cal Poly

Pomona. The subject is the design and analysis of adaptive methods for use with the finite element exterior calculus.

Holst is Co-PI on the project *Adaptive Radiotherapy Based on High Performance Computing* funded by the Department of Energy, Lawrence Livermore National Laboratory, and the University of California, 2009-2012. Collaborators include Steve Jian (UCSD Medical School), A. Majumdar (SDSC), and D.J. Choi (SDSC). The subject is realtime solution of coupled reaction-diffusion systems and the Boltzmann transport equation using a combination of parallel algorithms for partial differential equations, high-speed communication networks, and cluster computers.

Holst is Co-PI on the project *Scalable Adaptive Multilevel Solvers for Multiphysics Problems*, funded by the Department of Energy. The subject is the design and analysis of determinstic algorithms for use in physical simulation based on multilevel technologies.

Holst is Co-PI on the project *Applications of Quantum Computing in Aerospace Science and Engineering*, funded by the AirForce Office of Scientific Research. The subject is the design and analysis of quantum algorithms for use in physical simulation.

Holst is Co-PI and Core 1A lead on the project *National Biomedical Computation Resource (NBCR)* funded by the National Institutes of Health, 2009-2014. Collaborators include Andrew McCammon (UCSD Chemistry), Andrew McCulloch (UCSD Bioengineering), Mark Ellisman (UCSD Medical School), and Peter Arzberger (SDSC). The subject is multiscale modeling frameworks and adaptive finite element methods for complex multiscale and multiphysics problems arising in biomedical science.

Holst is Senior Scientist and founding member of the *NSF Physics Frontier Center for Theoretical Biological Physics (CTBP)*, funded by the National Science Foundation. Collaborators include Jose' Onuchic (UCSD Physics), Andrew McCammon (UCSD Chemistry), and Andy Kummel (UCSD Chemistry). The subject is multiscale modeling frameworks and adaptive finite element methods for complex multiscale and multiphysics problems arising in biophysics.

5.9. Transitions to technology applications

We report on current interactions with industry.

Estep was a Co-Principal Investigator in the Tech X, Inc. project *Framework Application for Core-Edge Transport Simulations (FACETS)*, funded by the Office of Advanced Scientific Computing Research and Office of Fusion Energy Sciences, Department of Energy. Estep's responsibilities include development and analysis of numerical solution methods for coupled core-edge fusion simulations. Algorithms developed in this program will be implemented into the FACETS high performance framework.

Estep was a subcontract in Phase II project for *Multiscale Design Systems, LLC* (Principal Officer: Jacob Fish, Rensselaer Polytechnic Institute) for the Air Force SBIR/STTR program. Estep's responsibilities include development of multiscale operator decomposition numerical methods and numerical methods for error estimation, uncertainty quantification and inverse problems for parameter identification for multiscale multiphysics models of hygro-thermo-mechano-oxidation-fatigue in polymer matrix composites used in aircraft applications. Algorithms developed in this program will be implemented into the Multiscale Design System for Continuum (MDS-C) and the Multiscale Design System for Discrete or atomistic medium (MDS-D) software packages.

Holst is collaborating with Eric Bylaska at Pacific Northwest National Laboratory on the incorporation of the Finite Element Toolkit (FETK, developed and maintained by the Holst Group) into several density functional modeling packages based at PNNL.

6. Distribution List

The final report is distributed to

DTIC-OCP
8725 John J Kingman Road, Suite 0944
Fort Belvoir, VA 22060-6218

DTRA
ATTN: NTRM
6200 Meade Road
Fort Belvoir, VA 22060-5264

DTRA
ATTN: BE-BCR
6200 Meade Road
Fort Belvoir, VA 22060-5264

DTRA
ATTN: BE-BLMI
6200 Meade Road
Fort Belvoir, VA 22060-5264

SIAM J. SCI. COMPUT.
Vol. 31, No. 4, pp. 2935–2959

NONPARAMETRIC DENSITY ESTIMATION FOR RANDOMLY PERTURBED ELLIPTIC PROBLEMS I: COMPUTATIONAL METHODS, A POSTERIORI ANALYSIS, AND ADAPTIVE ERROR CONTROL*

D. ESTEP†, A. MÅLQVIST‡, AND S. TAVENER§

Abstract. We consider the nonparametric density estimation problem for a quantity of interest computed from solutions of an elliptic partial differential equation with randomly perturbed coefficients and data. Our particular interest are problems for which limited knowledge of the random perturbations are known. We derive an efficient method for computing samples and generating an approximate probability distribution based on Lion's domain decomposition method and the Neumann series. We then derive an a posteriori error estimate for the computed probability distribution reflecting all sources of deterministic and statistical errors. Finally, we develop an adaptive error control algorithm based on the a posteriori estimate.

Key words. a posteriori error analysis, adjoint problem, density estimation, domain decomposition, elliptic problem, Neumann series, nonparametric density estimation, random perturbation, sensitivity analysis

AMS subject classifications. 65N15, 65N30, 65N55, 65C05

DOI. 10.1137/080731670

1. Introduction.

The practical application of differential equations to model physical phenomena presents problems in both computational mathematics and statistics. The mathematical issues arise because of the need to compute approximate solutions of difficult problems, while statistics arises because of the need to incorporate experimental data and model uncertainty. The consequence is that significant error and uncertainty attend any computed information from a model applied to a concrete situation. The problem of quantifying that error and uncertainty is critically important.

We consider the nonparametric density estimation problem for a quantity of interest computed from the solutions of an elliptic partial differential equation with randomly perturbed coefficients and data. The ideal problem is to compute a quantity

*Received by the editors July 20, 2008; accepted for publication (in revised form) May 18, 2009; published electronically July 3, 2009.

http://www.siam.org/journals/sisc/31-4/73167.html

†Department of Mathematics and Department of Statistics, Colorado State University, Fort Collins, CO 80523 (estep@math.colostate.edu). This author's work was supported by the Department of Energy (DE-FG02-04ER25620, DE-FG02-05ER25699, and DE-FC02-07ER54909), Lawrence Livermore National Laboratory (B573139), the National Aeronautics and Space Administration (NNG04GH63G), the National Science Foundation (DMS-0107832, DMS-0715135, DGE-0221595003, MSPA-CSE-0434354, and ECCS-0700559), Idaho National Laboratory (00069249), and the Sandia Corporation (PO299784).

‡Department of Information Technology, Uppsala University, SE-751 05 Uppsala, Sweden (axel.malqvist@it.uu.se). This author's work was supported by the Department of Energy DE-FG02-04ER25620.

§Department of Mathematics, Colorado State University, Fort Collins, CO 80523 (tavener@math.colostate.edu). This author's work was supported by the Department of Energy (DE-FG02-04ER25620).

of interest $Q(U)$, expressed as a linear functional, of the solution U of

$$\begin{cases} -\nabla \cdot (\mathbb{A}(x)\nabla U) = G(x), & x \in \Omega, \\ U = 0, & x \in \partial\Omega, \end{cases} \tag{1.1}$$

where Ω is a convex polygonal domain with boundary $\partial\Omega$, and $\mathbb{A}(x)$ and $G(x)$ are stochastic functions that vary randomly according to some given probability structure. The problem (1.1) is interpreted to hold almost surely (a.s.), i.e., with probability 1. Under suitable assumptions, e.g., $\mathbb{A}$ and G are uniformly bounded and have piecewise smooth dependence on their inputs (a.s.) with continuous and bounded covariance functions and $\mathbb{A}$ is uniformly coercive, $Q(U)$ is a random variable. The density estimation problem is as follows: Given probability distributions describing the stochastic nature of $\mathbb{A}$ and G, determine the probability distribution of Q. The approach we use extends to problems with more general Dirichlet or Robin boundary conditions in which the data for the boundary conditions are randomly perturbed as well as problems with more general elliptic operators in a straightforward way.

The parametric density estimation problem assumes that the output distribution is one of the standard distributions so that the problem involves determining values for the parameters defining the distribution. The nonparametric density estimation problem is relevant when the output distribution is unknown and/or complicated, e.g., multimodal. In this case, we seek to compute an approximate distribution for the output random variable using sample solutions of the problem. A limited version of this problem is to seek only to compute one or two moments, e.g., the expected value. However, this is of limited utility when the output distribution is complicated, as it tends to be for outputs computed from (1.1) under general conditions.

Nonparametric density estimation problems are generally approached using a Monte Carlo sampling method. Samples $\{\mathbb{A}^n, G^n\}$ are drawn from their distributions, solutions $\{U^n\}$ are computed to produce samples $\{Q(U^n)\}$, and the output distribution is approximated using a binning strategy coupled with smoothing. This ideal density estimation problem poses several computational issues, including the following.

1. We have only limited information about the stochastic nature of $\mathbb{A}$ and G.
2. We can compute only a finite number $\mathcal{N}$ of sample solutions.
3. The solution of (1.1) has to be computed numerically, which is both expensive and leads to significant variation in the numerical error as the coefficients and data vary.
4. The output distribution is an approximation affected by the binning and smoothing strategies.

In this paper, we consider the first three issues. Our goals are to construct an efficient numerical method for approximating the cumulative density function for the output distribution and to derive computable a posteriori error estimates that account for the significant effects of error and uncertainty in the approximation. We fully develop the adaptive algorithm, extend the analysis to include adaptive modeling, and test the algorithm on several problems in [5]. In [3], we present convergence proofs for the method described in this paper. There are many papers addressing the fourth issue in the statistics literature, e.g., kernel density estimation.

Our main goal is to treat the effects of stochastic variation in the diffusion coefficient $\mathbb{A}$. The treatment of a problem in which just the right-hand side and data vary stochastically is somewhat easier because there is just one differential operator to be inverted. When the elliptic coefficient varies stochastically, we are dealing with

a family of differential operators. We include a brief treatment of stochastic variation in the right-hand side and data to be complete.

1.1. Some notation. The notation is cumbersome since we are dealing with two discretizations: solution of the differential equation and approximation of a probability distribution by finite sampling. Generally, capital letters denote random variables, or samples, and lowercase letters represent deterministic variables or functions. When this assignment is violated, we use italics to denote deterministic quantities. We let $\Omega \subset \mathbf{R}^d$, $d = 2, 3$, denote the piecewise polygonal computational domain with boundary $\partial\Omega$. For an arbitrary domain $\omega \subset \Omega$ we denote the $L^2(\omega)$ scalar product by $(v, w)_\omega = \int_\omega vw\,dx$ in the domain and $\langle v, w\rangle_{\partial\omega} = \int_{\partial\omega} vw\,ds$ on the boundary, with associated norms $\|\ \|_\omega$ and $|\ |_\omega$. When $\omega = \Omega$, we drop the index in the scalar products. We let $\mathcal{H}^s(\omega)$ denote the standard Sobolev space of smoothness s for $s \geq 0$. In particular, $\mathcal{H}_0^1(\Omega)$ denotes the space of functions in $\mathcal{H}^1(\Omega)$ for which the trace is 0 on the boundary. If $\omega = \Omega$, we drop ω, and also if $s = 0$, we drop s; i.e., $\|\cdot\|$ denotes the $L^2(\Omega)$-norm.

We assume that any random vector X is associated with a probability space $(\Lambda, \mathcal{B}, P)$ in the usual way. We let $\{X^n, n = 1, \ldots, \mathcal{N}\}$ denote a collection of samples. We assume it is understood how to draw these samples. We let $E(X)$ denote the expected value, $\mathrm{Var}(X)$ denote the variance, and $F(t) = P(X < t)$ denote the cumulative distribution function. We compute approximate cumulative distribution functions in order to determine the probability distribution of a random variable.

1.2. A modeling assumption. The first step in developing a numerical method for the density estimation problem is to characterize the stochastic nature of the random variations affecting the problem. We assume that the stochastic diffusion coefficient can be written

$$\mathbb{A} = a + A,$$

where the uniformly coercive, bounded deterministic function a may have multiscale behavior and A describes a relatively small stochastic perturbation. Specifically, we assume that $a(x) \geq a_0 > 0$ for $x \in \Omega$ and that $|A(x)| \leq \delta a(x)$ for some $0 < \delta < 1$.

As a modeling assumption, we assume that A is a piecewise constant function with random coefficients. Specifically, we let $\mathcal{K}$ be a finite polygonal partition of Ω, where $\Omega = \cup_{\kappa\in\mathcal{K}}\kappa$ and κ_1 and κ_2 either are disjoint or intersect only along a common boundary when $\kappa_1 \neq \kappa_2$. We let χ_κ denote the characteristic function for the set $\kappa \in \mathcal{K}$. We assume that

$$A(x) = \sum_{\kappa\in\mathcal{K}} A^\kappa\, \chi_\kappa(x), \quad x \in \Omega, \tag{1.2}$$

where (A^κ) is a random vector and each coefficient A^κ is associated with a given probability distribution. We illustrate such a representation in Figure 1.1. Improving the model under this assumption requires choosing a finer partition and taking more measurements A^κ; see [5].

Note that we do *not* assume that the coefficients of A are independent and/or uncorrelated. We assume only that it is possible to draw samples of the values. We denote a finite set of samples by $\{A^{n,\kappa}, n = 1, \ldots, \mathcal{N}\}$. There are a few situations in which this is reasonable; e.g., see the following.

- There may be a component of the diffusion coefficient and/or its error that can be determined experimentally only at a relatively small set of points in

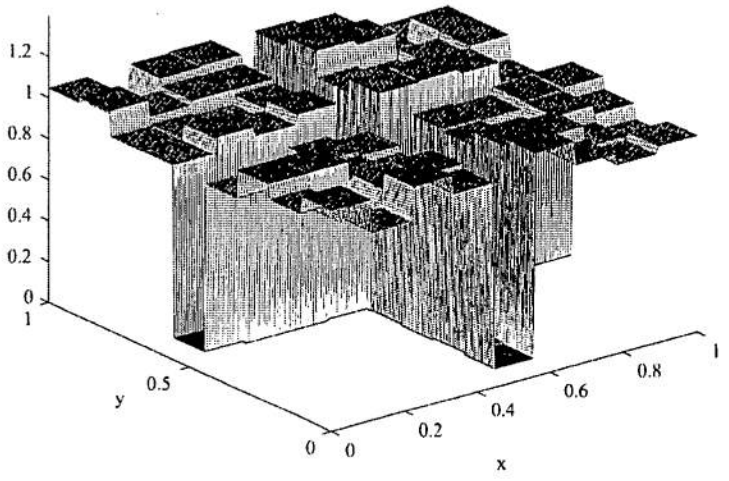


FIG. 1.1. *Illustration of the modeling assumption* (1.2). *The unit square is partitioned into* 9×9 *identical squares. The diffusion coefficient* a *is* 0.1 *on the "cross"-shaped domain centered at the origin and* 1 *elsewhere. The random perturbations are uniform on the interval determined by* $\pm 10\%$ *of the value of* a.

the domain Ω. For example, consider the SPE10 comparative oil reservoir simulation project description in [6]. In the absence of more information, it is natural to build a piecewise constant description of the measured information.

- We may assume or have knowledge of global distribution governing the random perturbation and simply represent realizations of the perturbation as piecewise constant functions with respect to a given domain partition.

We show below that assuming a spatially localized, piecewise constant description for A provides the possibility of devising a very efficient density estimation algorithm. In [3], we treat the situation in which A is a piecewise polynomial function, and in particular A may be continuous. An alternative, powerful approach to describe random behavior is based on the use of Karhunen–Loéve, polynomial chaos, or other orthogonal expansions of the random vectors [7, 1], which provides a spatially global representation. However, this approach requires detailed knowledge of the probability distributions for the input variables that is often not available.

2. The case of a randomly perturbed diffusion coefficient. We begin by studying the Poisson equation with a randomly perturbed diffusion coefficient. We let $U \in \mathcal{H}_0^1(\Omega)$ (a.s.) solve

$$\begin{cases} -\nabla \cdot \mathbb{A}\nabla U = f, & x \in \Omega, \\ U = 0, & x \text{ in } \partial\Omega, \end{cases} \tag{2.1}$$

where $f \in L^2(\Omega)$ is a given deterministic function and $\mathbb{A} = a + A$ satisfies the conditions described in section 1.2. We construct an efficient numerical method for computing sample solutions and then provide an a posteriori analysis of the error of the method.

2.1. More notation. We use the finite element method to compute numerical solutions. First, some general notation is as follows: Let $\mathcal{T}_h = \{\tau\}$ be a quasi-uniform partition into elements that $\cup\tau = \Omega$. Associated with $\mathcal{T}_h$, we define the discrete finite

element space $\mathcal{V}_h$ consisting of continuous, piecewise linear functions on $\mathcal{T}$ satisfying Dirichlet boundary conditions, with mesh size function $h_\tau = \operatorname{diam}(\tau)$ for $x \in \tau$ and $h = \max_{\tau \in \mathcal{T}_h} h_\tau$. In some situations, we use a more accurate finite element space $\mathcal{V}_{\tilde{h}}$ either comprising the space of continuous, piecewise quadratic functions $\mathcal{V}_h^2$ or involving a refinement $\mathcal{T}_{\tilde{h}}$ of $\mathcal{T}_h$, where $\tilde{h} \ll h$.

Our approach uses Lion's nonoverlapping domain decomposition method [9, 8]. Again, some general notation is as follows: We let $\{\Omega_d,\ d = 1, \ldots, \mathcal{D}\}$ be a decomposition of Ω into a finite set of nonoverlapping polygonal subdomains with $\cup\Omega_d = \Omega$. We denote the boundaries by $\partial\Omega_d$ and outward normals by $\boldsymbol{n}_d$. For a function $\mathbb{A}^n$ on Ω, $\mathbb{A}^{n,d}$ means $\mathbb{A}^n$ restricted to Ω_d. For $d = 1, \ldots, \mathcal{D}$, d' denotes the set of indices in $\{1, 2, \ldots, \mathcal{D}\} \setminus \{d\}$ for which the corresponding domains $\Omega_{d'}$ share a common boundary with Ω_d. The method is iterative, so for a function $\mathcal{U}$ involved in the iteration, $\mathcal{U}_i$ denotes the value at the ith iteration. Let $\mathbb{A}^n = a + A^n$ be a particular sample of the diffusion coefficient with corresponding solution U^n. We let $\{U_0^{n,d}, d = 1, \ldots, \mathcal{D}\}$ denote a set of initial guesses for solutions in the subdomains.

2.2. The computational method. Returning to (2.1), we assume that the finite element discretization $\mathcal{T}_h$ is obtained by refinement of $\mathcal{K}$ associated with the modeling assumption (1.2) on A. This is natural when the diffusion coefficient a and the data vary on a scale finer than the partition $\mathcal{K}$.

Given the initial conditions, for each $i \geq 1$, we numerically solve the $\mathcal{D}$ problems

$$\begin{cases} -\nabla \cdot \mathbb{A}^n \nabla U_i^{n,d} = f, & x \in \Omega_d, \\ U_i^{n,d} = 0, & x \in \partial\Omega_d \cap \partial\Omega, \\ \frac{1}{\lambda} U_i^{n,d} + \boldsymbol{n}_d \cdot \mathbb{A}^n \nabla U_i^{n,d} = \frac{1}{\lambda} U_{i-1}^{n,\tilde{d}} - \boldsymbol{n}_{\tilde{d}} \cdot \mathbb{A}^n \nabla U_{i-1}^{n,\tilde{d}}, & x \in \partial\Omega_d \cap \partial\Omega_{\tilde{d}}, \quad \tilde{d} \in d', \end{cases}$$

where the parameter $\lambda \in \mathbf{R}$ is chosen to minimize the number of iterations. In practice, we compute $\mathcal{I}$ iterations. Note that the problems can be solved independently.

To discretize, we let $\mathcal{V}_{h,d} \subset \mathcal{H}^1_{0,\partial\Omega}(\Omega_d)$ be a finite element approximation space corresponding to the mesh $\mathcal{T}_d$ on Ω_d, where

$$\mathcal{H}^1_{0,\partial\Omega}(\Omega_d) = \{v \in \mathcal{H}^1(\Omega_d) :\ v|_{\partial\Omega_D \cap \partial\Omega} = 0\}.$$

We let $(\cdot,\cdot)_d$ denote the $L^2(\Omega_d)$ scalar product, $\langle\cdot,\cdot\rangle_d$ denote the $L^2(\partial\Omega_d)$ scalar product, and $\langle\cdot,\cdot\rangle_{d\cap\tilde{d}}$ denote the $L^2(\partial\Omega_d \cap \partial\Omega_{\tilde{d}})$ scalar product for $\tilde{d} \in d'$. The first two inner products are associated with norms $\|\ \|_d$ and $|\ |_d$, respectively. For each $i \geq 1$, we compute $U_i^{n,d} \in \mathcal{V}_{h,d}$, $d = 1, \ldots, \mathcal{D}$, solving

$$\begin{aligned} (2.2)\quad & (\mathbb{A}^n \nabla U_i^{n,d}, \nabla v)_d + \frac{1}{\lambda} \langle U_i^{n,d}, v \rangle_d \\ & = (f, v)_d + \sum_{\tilde{d} \in d'} \left(\frac{1}{\lambda} \langle U_{i-1}^{n,\tilde{d}}, v \rangle_{d\cap\tilde{d}} - \langle \boldsymbol{n}_{\tilde{d}} \cdot \mathbb{A}^n \nabla U_{i-1}^{n,\tilde{d}}, v \rangle_{d\cap\tilde{d}} \right) \quad \text{for all } v \in \mathcal{V}_{h,d}. \end{aligned}$$

It is convenient to use the matrix form of (2.2). We let $\{\varphi_m^d, m = 1, \ldots, n_d\}$ be the finite element basis functions for the space $\mathcal{V}_{h,d}$, $d = 1, \ldots, \mathcal{D}$. We let $\vec{U}_i^{n,d}$ denote the vector of basis coefficients of $U_i^{n,d}$ with respect to $\{\varphi_m^d\}$. On each domain Ω_d,

$$(\mathbf{k}^{a,d} + \mathbf{k}^{n,d}) \vec{U}_i^{n,d} = \vec{b}^d(f) + \vec{b}^{n,d}(\mathbb{A}^n, U_{i-1}^{n,d'}),$$

where

$$
\begin{aligned}
(\mathbf{k}^{a,d})_{lk} &= (a\nabla\varphi_l^d, \nabla\varphi_k^d)_d + \frac{1}{\lambda}\langle\varphi_l^d, \varphi_k^d\rangle_d,\\
(\mathbf{k}^{n,d})_{lk} &= (A^{n,d}\nabla\varphi_l^d, \nabla\varphi_k^d)_d,\\
(\vec{b}^d)_k &= (f, \varphi_k^d)_d,\\
(\vec{b}^{n,d})_k &= \sum_{\tilde{d}\in d'}\left(\frac{1}{\lambda}\langle U_{i-1}^{n,\tilde{d}}, \varphi_k^d\rangle_{d\cap\tilde{d}} - \langle n_{\tilde{d}}\cdot\mathbb{A}^n\nabla U_{i-1}^{n,\tilde{d}}, \varphi_k^d\rangle_{d\cap\tilde{d}}\right)
\end{aligned}
$$

for $1 \le l$ and $k \le n_d$. We abuse notation mildly by denoting the dependence of the data $\vec{b}^{n,d}(\mathbb{A}^n, U_{i-1}^{n,d'})$ on the values of $U_{i-1}^{n,\tilde{d}}$ for $\tilde{d} \in d'$. We summarize this approach in Algorithm 1.

Algorithm 1. MONTE CARLO DOMAIN DECOMPOSITION FINITE ELEMENT METHOD

```
for n = 1,...,𝒩 (number of samples) do
  for i = 1,...,ℐ (number of iterations) do
    for d = 1,...,D (number of domains) do
      Solve
      Solve (k^{a,d} + k^{n,d}) U_i^{n,d} = b^d(f) + b^{n,d}(𝔸^n, U_{i-1}^{n,d'}).
    end for
  end for
end for
```

Unfortunately, this algorithm is expensive for a large number of realizations since each solution U^n requires the solution of a discrete set of equations. To construct a more efficient method, we impose a restriction on the domains in the decomposition. We assume that each domain Ω_d is contained in a domain κ in the partition $\mathcal{K}$ used in the modeling assumption (1.2). This implies that the random perturbation $A^{n,d}$ is constant on each Ω_d; i.e., it is a random number. Consequently, the matrix $\mathbf{k}^{n,d}$ has coefficients

$$
(\mathbf{k}^{n,d})_{lk} = (A^{n,d}\nabla\varphi_l^d, \nabla\varphi_k^d)_d = A^{n,d}(\nabla\varphi_l^d, \nabla\varphi_k^d)_d = A^{n,d}(\mathbf{k}^d)_{lk},
$$

where $\mathbf{k}^d$ is the standard stiffness matrix with coefficients $(\mathbf{k}^d)_{lk} = (\nabla\varphi_l^d, \nabla\varphi_k^d)_d$. We now use the fact that $A^{n,d}$ is relatively small to motivate the introduction of the Neumann series. Formally, the Neumann series for the inverse of a perturbation of the identity matrix provides

$$
\begin{aligned}
\left(\mathbf{k}^{a,d} + A^{n,d}\mathbf{k}^d\right)^{-1} &= \left(\mathbf{k}^{a,d}\left(\mathbf{id} + A^{n,d}(\mathbf{k}^{a,d})^{-1}\mathbf{k}^d\right)\right)^{-1}\\
&= \left(\mathbf{id} + A^{n,d}(\mathbf{k}^{a,d})^{-1}\mathbf{k}^d\right)^{-1}(\mathbf{k}^{a,d})^{-1}\\
&= \sum_{p=0}^{\infty}(-A^{n,d})^p\left((\mathbf{k}^{a,d})^{-1}\mathbf{k}^d\right)^p(\mathbf{k}^{a,d})^{-1},
\end{aligned}
$$

where $\mathbf{id}$ is the identity matrix. We compute only $\mathcal{P}$ terms in the Neumann expansion to generate the approximation

$$
\vec{U}_{\mathcal{P},i}^{n,d} = \sum_{p=0}^{\mathcal{P}-1}\left((-A^{n,d})^p((\mathbf{k}^{a,d})^{-1}\mathbf{k}^d)^p\right)(\mathbf{k}^{a,d})^{-1}\left(\vec{b}^d(f) + \vec{b}^{n,d}(\mathbb{A}^n, U_{\mathcal{P},i-1}^{n,d'})\right). \tag{2.3}
$$

We discuss the convergence as $\mathcal{P} \to \infty$ in detail below.

Note that $\vec{b}^{n,d}$ is nonzero only at boundary nodes. If $\mathcal{W}_{h,d}$ denotes the set of vectors determined by the finite element basis functions associated with the boundary nodes on Ω_d, then $\vec{b}^{n,d}$ is in the span of $\mathcal{W}_{h,d}$. We can precompute

$$((\mathbf{k}^{a,d})^{-1}\mathbf{k}^d)^p(\mathbf{k}^{a,d})^{-1}\mathcal{W}_{h,d}$$

efficiently, e.g., using Gaussian elimination. This computation is independent of n.

Algorithm 2. MONTE CARLO DOMAIN DECOMPOSITION FINITE ELEMENT METHOD USING A TRUNCATED NEUMANN SERIES

for $d = 1, \ldots, D$ (number of domains) **do**
  for $p = 1, \ldots, \mathcal{P}$ (number of terms) **do**
    Compute $\vec{y} = \left((\mathbf{k}^{a,d})^{-1}\mathbf{k}^d\right)^p(\mathbf{k}^{a,d})^{-1}\vec{b}^d(f)$
    Compute $\mathbf{y}^p = \left((\mathbf{k}^{a,d})^{-1}\mathbf{k}^d\right)^p(\mathbf{k}^{a,d})^{-1}\mathcal{W}_{h,d}$
  end for
end for
for $i = 1, \ldots, \mathcal{I}$ (number of iterations) **do**
  for $d = 1, \ldots, D$ (number of domains) **do**
    for $p = 1, \ldots, \mathcal{P}$ (number of terms) **do**
      for $n = 1, \ldots, \mathcal{N}$ (number of samples) **do**
        Compute $\vec{U}^{n,d}_{\mathcal{P},i} = \sum_{p=0}^{\mathcal{P}-1}(-A^{n,d})^p\left(\mathbf{y}^p\vec{b}^{n,d}(\mathbb{A}^n, U^{n,d'}_{\mathcal{P},i-1}) + \vec{y}\right)$
      end for
    end for
  end for
end for

Combining this with Algorithm 1, we obtain the computational method given in Algorithm 2. We let $U^{n,d}_{\mathcal{P},\mathcal{I}}$ denote the finite element functions determined by $\vec{U}^{n,d}_{\mathcal{P},\mathcal{I}}$ for $n = 1, \ldots, \mathcal{N}$ and $d = 1, \ldots, \mathcal{D}$. We let $U^n_{\mathcal{P},\mathcal{I}}$ denote the finite element function which is equal to $U^{n,d}_{\mathcal{P},\mathcal{I}}$ on Ω_d.

Remark 2.1. Note that the number of linear systems that have to be solved in Algorithm 2 is independent of $\mathcal{N}$. Hence, there is potential for enormous savings when the number of samples is large.

2.3. Convergence of the Neumann series approximation. It is crucial for the method that the Neumann series converges. The following theorem shows convergence under the reasonable assumption that the random perturbations $A^{n,d}$ to the diffusion coefficient are smaller than the coefficient. We let $|||v|||_d^2 = \|\nabla v\|_d^2 + \epsilon|v|_d^2$ for some $\epsilon > 0$. We define the matrices $\mathbf{c}^{n,d} = -A^{n,d}(\mathbf{k}^{a,d})^{-1}\mathbf{k}^d$ and denote the corresponding operators on the finite element spaces by $c^{n,d} : \mathcal{V}_{h,d} \to \mathcal{V}_{h,d}$.

THEOREM 2.1. *If* $\eta = |\max\{A^{n,d}\}/a_0| < 1$, *then*

$$\text{(2.4)} \quad \text{(a)} \qquad \left(\mathbf{id} - \mathbf{c}^{n,d}\right)^{-1} = \sum_{p=0}^{\infty}\left(\mathbf{c}^{n,d}\right)^p,$$

$$\text{(2.5)} \quad \text{(b)} \qquad \left|\left|\left|\left(\left(1 - c^{n,d}\right)^{-1} - \sum_{p=0}^{\mathcal{P}-1}\left(c^{n,d}\right)^p\right)v\right|\right|\right|_d \le \frac{\eta^{\mathcal{P}}}{1-\eta^{\mathcal{P}}}\left|\left|\left|\sum_{p=0}^{\mathcal{P}-1}\left(c^{n,d}\right)^p v\right|\right|\right|_d$$

for any $v \in \mathcal{V}_{h,d}$.

Proof. Let $z = c^{n,d}w$ for an arbitrary $w \in \mathcal{V}_{h,d}$. From the definition of $c^{n,d}$, $z \in \mathcal{V}_{h,d}$ satisfies

$$(a\nabla z, \nabla v)_d + \frac{1}{\lambda}(z, v\rangle_d = -A^{n,d}(\nabla w, \nabla v)_d \tag{2.6}$$

for all $v \in \mathcal{V}_{h,d}$. Choosing $v = z$ in (2.6) and using the Cauchy–Schwarz inequality yields

$$\|\nabla z\|_d^2 + \frac{1}{a_0\lambda}|z|_d^2 \leq \eta\|\nabla w\|_d\|\nabla z\|_{\Omega_d}.$$

Choosing $\epsilon < 2/(\lambda a_0)$ in the definition of the norm $|||v|||_d^2 = \|\nabla v\|_d^2 + \epsilon|v|_d^2$ and making standard estimates gives

$$|||z|||_d^2 \leq \eta^2\|\nabla w\|_d^2 \leq \eta^2|||w|||_d^2.$$

By induction,

$$\left|\left|\left|\left(c^{n,d}\right)^p w\right|\right|\right|_d \leq \eta^p|||w|||_d. \tag{2.7}$$

In particular, $\left(c^{n,d}\right)^p \to 0$ as $p \to \infty$.

We take the limit as $\mathcal{P}$ tends to infinity in the identity

$$\mathbf{id} - \left(\mathbf{c}^{n,d}\right)^{\mathcal{P}} = \left(\mathbf{id} - \mathbf{c}^{n,d}\right)\sum_{p=0}^{\mathcal{P}-1}\left(\mathbf{c}^{n,d}\right)^p$$

to obtain (2.4).

In order to prove (b) we note that

$$\left(\mathbf{id} - \mathbf{c}^{n,d}\right)^{-1} - \sum_{p=0}^{\mathcal{P}-1}\left(\mathbf{c}^{n,d}\right)^p = \sum_{p=\mathcal{P}}^{\infty}\left(\mathbf{c}^{n,d}\right)^p = \left(\mathbf{c}^{n,d}\right)^{\mathcal{P}}\left(\mathbf{id} - \mathbf{c}^{n,d}\right)^{-1}.$$

In the finite element context, for $v \in \mathcal{V}_{h,d}$,

$$\begin{aligned}
&\left|\left|\left|\left(1 - c^{n,d}\right)^{-1}v - \sum_{p=0}^{\mathcal{P}-1}\left(c^{n,d}\right)^p v\right|\right|\right|_d \\
&\qquad = \left|\left|\left|\left(c^{n,d}\right)^{\mathcal{P}}\left(1 - c^{n,d}\right)^{-1}v\right|\right|\right|_d \leq \eta^{\mathcal{P}}\left|\left|\left|\left(1 - c^{n,d}\right)^{-1}v\right|\right|\right|_d \\
&\qquad \leq \eta^{\mathcal{P}}\left|\left|\left|\sum_{p=0}^{\mathcal{P}-1}\left(c^{n,d}\right)^p v\right|\right|\right|_d + \eta^{\mathcal{P}}\left|\left|\left|\left(1 - c^{n,d}\right)^{-1}v - \sum_{p=0}^{\mathcal{P}-1}\left(c^{n,d}\right)^p v\right|\right|\right|_d.
\end{aligned}$$

The theorem follows immediately. □

2.4. A numerical example. We present a numerical example that illustrates the convergence properties of the proposed method. Below, we derive an a posteriori estimate for the contributions to the error of the approximation and develop an adaptive algorithm that provides the means to balance these contributions efficiently.

In this example, we partition the unit square into 9×9 equal square subdomains for the domain decomposition algorithm. The coefficient $\mathbb{A}^n = a + A^n$, where a and A^n are piecewise constant on the 9×9 subdomains. The diffusion coefficient a is equal to 1 except on a cross in the center of the domain Ω where it is equal to 0.1. The random perturbations are uniform on the interval determined by $\pm 10\%$ of the value of a. We illustrate a typical sample in Figure 1.1. The data is $f \equiv 1$. We estimate the error in the quantity of interest which is the average of the solution.

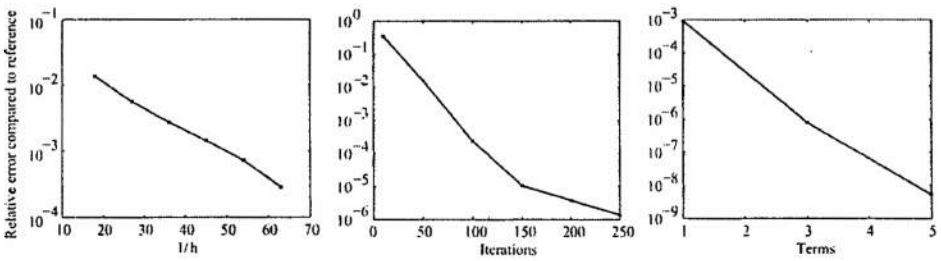


FIG. 2.1. *Left: convergence in mesh size as the numbers of domain decomposition iterations and terms in the Neumann series are held constant. Middle: convergence with respect to the number of domain decomposition iterations as the mesh size and number of terms in the Neumann series are held constant. Right: convergence with respect to the number of terms in the Neumann series as the mesh size and number of domain decomposition iterations are held constant.*

2.4.1. Convergence for a single realization. First, we consider a single sample A^n in order to focus on the convergence with respect to mesh size, number of iterations, and number of terms. In particular, we study how the accuracy of a computed linear functional of the solution $(U^{1,d}_{\mathcal{P},\mathcal{I}}, \psi)$ depends on the three parameters h, $\mathcal{P}$, and $\mathcal{I}$. To compute approximate errors, we use a reference solution with $h = 1/72$, $\mathcal{I} = 300$, and $\mathcal{P} = 5$.

We start by letting $\mathcal{I} = 300$ and $\mathcal{P} = 5$, and let the number of elements in each direction $(1/h)$ vary from 18 to 72; i.e., the total number of nodes in the mesh varies from $(18 + 1)^2 = 361$ to $(72 + 1)^2 = 5329$. In Figure 2.1, we plot the relative error as the mesh size decreases. Next, we fix $1/h = 72$, keep $\mathcal{P} = 5$, and vary $\mathcal{I}$ from 10 to 300. In Figure 2.1, we plot the relative error as the number of iterations increase. Finally, we let $1/h = 72$, $\mathcal{I} = 300$, and $\mathcal{P}$ vary from 1, 3, and 5. Here, the reference solution is computed using $\mathcal{P} = 7$. We plot the results in Figure 2.1.

2.4.2. Convergence with respect to number of samples. Next, we fix the spatial discretization and experimentally investigate the accuracy in the cumulative distribution function as a function of the number of samples. Following the problem in section 2.4.1, we fix $h = 1/72$, $\mathcal{I} = 300$, and $\mathcal{P} = 5$, vary $\mathcal{N}$ from 30 to 480, and compute the cumulative distribution function $F_{\mathcal{N}}(t)$. We present the result in Figure 2.2. We observe that the distribution function becomes smoother as the number of samples increases and appears to converge.

To approximate the error as the samples increase, we compute a reference solution using $\mathcal{N} = 480$. We plot the errors in Figure 2.3. The error decreases significantly between $\mathcal{N} = 30$ and $\mathcal{N} = 240$, but the convergence is fairly slow.

2.5. A posteriori error analysis of sample values. We next derive an a posteriori error estimate for each sample linear functional value (U^n, ψ) [4, 2]. We introduce a corresponding adjoint problem

$$\begin{cases} -\nabla \cdot \mathbb{A}\nabla\Phi = \psi, & x \in \Omega, \\ \Phi = 0, & x \in \partial\Omega. \end{cases} \tag{2.8}$$

We compute $\mathcal{N}$ sample solutions $\{\Phi^n, n = 1, \ldots, \mathcal{N}\}$ of (2.8) corresponding to the samples $\{\mathbb{A}^n, n = 1, \ldots, \mathcal{N}\}$.

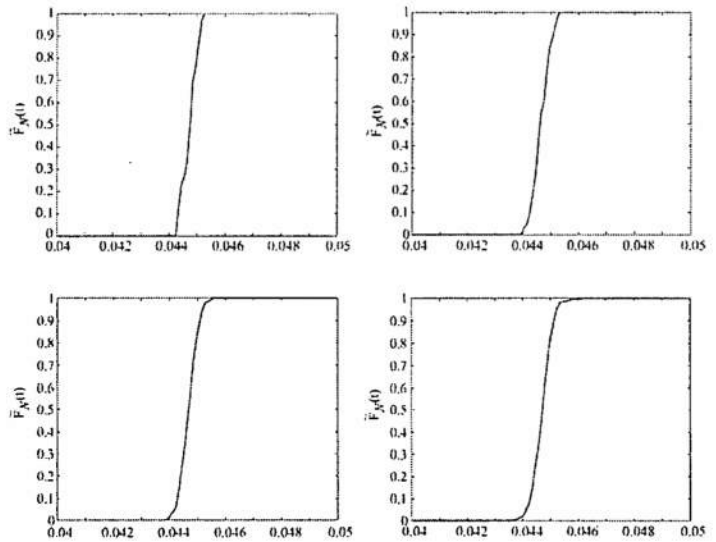


FIG. 2.2. *Convergence in the number of samples as the mesh, the number of domain decomposition iterations, and the number of terms in the Neumann series are held constant. Plots from left to right, top to bottom,* $\mathcal{N} = 30, 60, 120, 480$.

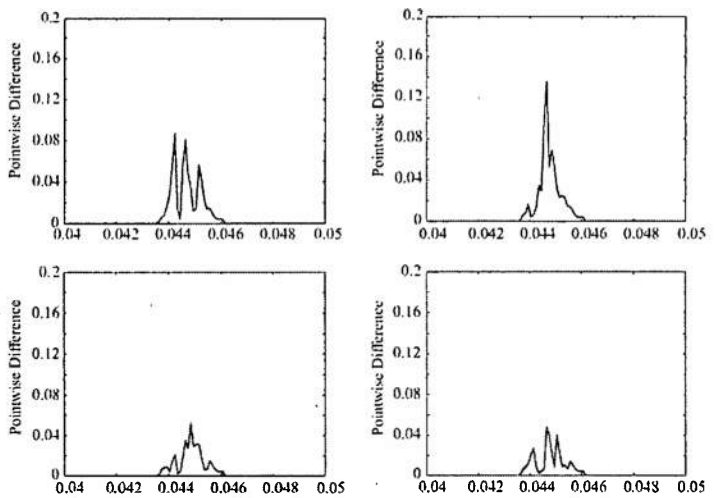


FIG. 2.3. *Error in the distribution function compared to a reference solution as the mesh, the number of iterations, and the number of terms are held constant. Plots from left to right, top to bottom,* $\mathcal{N} = 30, 60, 120, 240$.

To obtain computable estimates, we compute numerical solutions of (2.8) using Algorithm 2. We obtain numerical approximations $\{\Phi^{n,d}_{\tilde{\mathcal{P}},\tilde{\mathcal{I}}}, d = 1, \ldots, \mathcal{D}\}$ using a more accurate finite element discretization computed using either the space of continuous, piecewise quadratic functions $\mathcal{V}^2_h$ or a refinement $\mathcal{T}_{\tilde{h}}$ of $\mathcal{T}_h$, where $\tilde{h} \ll h$. We denote the approximation on Ω by $\Phi^n_{\tilde{\mathcal{P}},\tilde{\mathcal{I}}}$.

THEOREM 2.2. *For each* $n \in 1, \ldots, \mathcal{N}$,

$$\left|(U^n - U^n_{\mathcal{P},\mathcal{I}}, \psi)\right| \lessapprox \left|(f, \Phi^n_{\tilde{\mathcal{P}},\tilde{\mathcal{I}}}) - (\mathbb{A}^n \nabla U^n_{\mathcal{P},\mathcal{I}}, \nabla \Phi^n_{\tilde{\mathcal{P}},\tilde{\mathcal{I}}})\right| + \mathcal{R}(\tilde{h}, \tilde{\mathcal{P}}, \tilde{\mathcal{I}}), \tag{2.9}$$

where

$$\mathcal{R}(\tilde{h}, \tilde{\mathcal{P}}, \tilde{\mathcal{I}}) = \begin{cases} \mathbf{O}(h^3), & \mathcal{V}_{\tilde{h}} = \mathcal{V}_h^2, \\ \mathbf{O}(\tilde{h}^2), & \mathcal{T}_{\tilde{h}} \text{ is a refinement of } \mathcal{T}_h. \end{cases}$$

Proof. With Φ solving (2.8), the standard Green's function argument yields the representation

$$(U^n - U^n_{\mathcal{P},\mathcal{I}}, \psi) = (f, \Phi^n) - (\mathbb{A}^n \nabla U^n_{\mathcal{P},\mathcal{I}}, \nabla \Phi^n).$$

We write this as

$$\begin{aligned}(U^n - U^n_{\mathcal{P},\mathcal{I}}, \psi) = (f, \Phi^n_{\tilde{\mathcal{P}},\tilde{\mathcal{I}}}) - (\mathbb{A}^n \nabla U^n_{\mathcal{P},\mathcal{I}}, \nabla \Phi^n_{\tilde{\mathcal{P}},\tilde{\mathcal{I}}}) \\ + \left((f, \Phi^n - \Phi^n_{\tilde{\mathcal{P}},\tilde{\mathcal{I}}}) - (\mathbb{A}^n \nabla U^n_{\mathcal{P},\mathcal{I}}, \nabla(\Phi^n - \Phi^n_{\tilde{\mathcal{P}},\tilde{\mathcal{I}}}))\right)\end{aligned}$$

and define

$$\mathcal{R}(\tilde{h}, \tilde{\mathcal{P}}, \tilde{\mathcal{I}}) = (f, \Phi^n - \Phi^n_{\tilde{\mathcal{P}},\tilde{\mathcal{I}}}) - (\mathbb{A}^n \nabla U^n_{\mathcal{P},\mathcal{I}}, \nabla(\Phi^n - \Phi^n_{\tilde{\mathcal{P}},\tilde{\mathcal{I}}})). \tag{2.10}$$

We introduce auxiliary adjoint problems for the purpose of analysis. Let $\Upsilon^n \in \mathcal{V}$ solve

$$(\mathbb{A}^n \nabla \Upsilon^n, \nabla v) = (f, v) - (\mathbb{A}^n \nabla U^n_{\mathcal{P},\mathcal{I}}, v) \text{ for all } v \in \mathcal{V}, \tag{2.11}$$

corresponding to the quantity of interest $(f, \Phi) - (\mathbb{A}^n \nabla U^n_{\mathcal{P},\mathcal{I}}, \Phi^n)$. The standard Green's function argument yields

$$(f, \Phi^n - \Phi^n_{\tilde{\mathcal{P}},\tilde{\mathcal{I}}}) - (\mathbb{A}^n \nabla U^n_{\mathcal{P},\mathcal{I}}, \Phi^n - \Phi^n_{\tilde{\mathcal{P}},\tilde{\mathcal{I}}}) = (\Upsilon^n, \psi) - (\mathbb{A}^n \nabla \Upsilon^n, \nabla \Phi^n_{\tilde{\mathcal{P}},\tilde{\mathcal{I}}}).$$

Using $\Phi^n_{\infty,\tilde{\mathcal{I}}}$ to denote the approximate solution obtained by using the complete Neumann series (which is equivalent to finding the solution of the problem with the full diffusion coefficient), we have

$$\begin{aligned}(f, \Phi^n - \Phi^n_{\tilde{\mathcal{P}},\tilde{\mathcal{I}}}) - (\mathbb{A}^n \nabla U^n_{\mathcal{P},\mathcal{I}}, \Phi^n - \Phi^n_{\tilde{\mathcal{P}},\tilde{\mathcal{I}}}) = (\Upsilon^n, \psi) - (\mathbb{A}^n \nabla \Upsilon^n, \nabla \Phi^n_{\infty,\tilde{\mathcal{I}}}) \\ + (\mathbb{A}^n \nabla \Upsilon^n, \nabla(\Phi^n_{\tilde{\mathcal{P}},\tilde{\mathcal{I}}} - \Phi^n_{\infty,\tilde{\mathcal{I}}})).\end{aligned} \tag{2.12}$$

By Theorem 2.1, the third term on the right-hand side can be made arbitrarily small by taking $\mathcal{P}$ large. We can use Galerkin orthogonality on the first two terms on the right-hand side of (2.12) by introducing a projection $\pi_{\tilde{h}}$ into $\mathcal{V}_{\tilde{h}}$. Decomposing into a sum of integrals over elements and integrating by parts on each element, we have

$$\begin{aligned}(\Upsilon^n, \psi) - (\mathbb{A}^n \nabla \Upsilon^n, \nabla \Phi^n_{\infty,\tilde{\mathcal{I}}}) &= (\Upsilon^n - \pi_{\tilde{h}} \Upsilon^n, \psi) - (\mathbb{A}^n \nabla(\Upsilon^n - \pi_{\tilde{h}} \Upsilon^n), \nabla \Phi^n_{\infty,\tilde{\mathcal{I}}}) \\ &= \sum_{\tau \in \mathcal{T}_{\tilde{h}}} \Big((\Upsilon^n - \pi_{\tilde{h}} \Upsilon^n, \psi)_\tau + (\Upsilon^n - \pi_{\tilde{h}} \Upsilon^n, \nabla \cdot \mathbb{A}^n \nabla \Phi^n_{\infty,\tilde{\mathcal{I}}})_\tau \\ &\qquad + \langle \Upsilon^n - \pi_{\tilde{h}} \Upsilon^n, \mathbb{A}^n \partial_n \Phi^n_{\infty,\tilde{\mathcal{I}}} \rangle_{\partial\tau} \Big),\end{aligned}$$

where ∂_n denotes the normal derivative to $\partial\tau$. The standard argument involving interpolation estimates now yields the bounds in (2.10). ∎

2.5.1. Numerical example. We present a brief example illustrating the accuracy of the a posteriori estimate (2.9). We consider just one sample diffusion value $\mathbb{A}^1 = a + A^1$, with $a = 0.9$ and $A^1 = 0.1$ on the unit square. We compute the error in the average value by choosing $\psi = 1$. We set $f = 2 \cdot x(1-x) + 2 \cdot y(1-y)$ so that the exact solution is $U^1 = x(1-x) \cdot y(1-y)$ and $(U^1, \psi) = 1/36$.

We divide the computational domain into 8×8 equally sized squares on which we compute the numerical approximation to U^1 using Lion's domain decomposition algorithm with $\mathcal{I}$ iterations using the approximate local solver involving a truncated Neumann series of $\mathcal{P}$ terms. We let $h = 1/32$ so that each subdomain has discretization 5×5 nodes. To solve the adjoint problem, we use a refined mesh with $\tilde{h} = h/2$ and use $\gamma\mathcal{P}$ terms in the truncated Neumann series and $\gamma\mathcal{I}$ iterations in the domain decomposition algorithm, where $\gamma > 0$. To evaluate the accuracy of the estimate, we use the efficiency index

$$\eta^{\gamma}_{\mathcal{P},\mathcal{I}} = \frac{|(f, \Phi^1_{\gamma\mathcal{P},\gamma\mathcal{I}}) - (\mathbb{A}^1 \nabla U^1_{\mathcal{P},\mathcal{I}}, \nabla \Phi^1_{\gamma\mathcal{P},\gamma\mathcal{I}})|}{|(U^1 - U^1_{\mathcal{P},\mathcal{I}}, \psi)|}.$$

We start by letting $\gamma = 2$; i.e., we put a lot of effort into solving the adjoint solution. We present results for varying $\mathcal{I}$ and $\mathcal{P}$ in Table 2.1.

TABLE 2.1
Efficiency index results for $\gamma = 2$.

$\mathcal{P}$	$\mathcal{I}$	Ratio	$\mathcal{P}$	$\mathcal{I}$	Ratio
1	50	0.992	3	10	2.78
2	50	1.03	3	25	1.02
3	50	1.01	3	50	1.01

Next, we let $\gamma = 0.5$; i.e., we use much poorer resolution for the adjoint solution. We plot the results in Table 2.2.

TABLE 2.2
Efficiency index results for $\gamma = .5$.

$\mathcal{P}$	$\mathcal{I}$	Ratio	$\mathcal{P}$	$\mathcal{I}$	Ratio
1	50	0.908	3	10	8.21
2	50	1.00	3	25	1.08
3	50	0.933	3	50	0.933

The efficiency indexes are close to one except when the number of domain decomposition iterations for the adjoint problem is very low. In general, it appears that as long as the number of domain decomposition iterations is sufficiently large, the adjoint problem can be solved with rather poor resolution, yet we still obtain a reasonably accurate error estimate.

3. The case of random perturbation in data. For the sake of completeness, we treat the case in which the data G in (1.1) is randomly perturbed. It is straightforward to combine the cases of randomly perturbed diffusion coefficient and data. We present a fast method for computing samples of a linear functional of the solution given samples of the right-hand side data. It is straightforward to deal with a more general elliptic operator, so we let $U \in \mathcal{H}^1_0(\Omega)$ (a.s.) solve

$$a(U, v) = (G(x), v), \quad v \in \mathcal{H}^1_0(\Omega), \tag{3.1}$$

where

$$a(w, v) = (a\nabla w, \nabla v) + (b \cdot \nabla w, v) + (cw, v)$$

for $w, v \in \mathcal{H}_0^1(\Omega)$, $G(x) \in L^2(\Omega)$ (a.s.), $G(\cdot)$ has a continuous and bounded covariance function, and a, b, and c are deterministic functions chosen such that (3.1) has a unique weak solution in $\mathcal{H}_0^1(\Omega)$. In particular, $a(x) \geq a_0 > 0$ for all x. We let $\{G^n(x), n = 1, \ldots, \mathcal{N}\}$ denote a finite collection of samples.

3.1. Computational method. In the case of randomly perturbed right-hand side and data, we can use the method of Green's functions to construct an efficient method for density estimation. We introduce a deterministic adjoint problem. We let the quantity of interest be a linear functional $Q(v) = (v, \psi)$ determined by a function $\psi \in L^2(\Omega)$ and construct the corresponding adjoint problem for the generalized Green's function $\phi \in \mathcal{H}_0^1(\Omega)$,

$$a^*(\phi, v) = (\psi, v), \quad v \in \mathcal{H}_0^1(\Omega), \tag{3.2}$$

where

$$a^*(w, v) = (\nabla w, \nabla v) - (\nabla \cdot (bw), v) + (cw, v).$$

It immediately follows that

$$(U^n, \psi) = a^*(\phi, U^n) = a(U^n, \phi) = (G^n, \phi)$$

for $n = 1, \ldots, \mathcal{N}$. By linearity, we see that $E(U) \in \mathcal{H}_0^1(\Omega)$ solves

$$a(E(U), v) = (E(G^n), v), \quad v \in \mathcal{H}_0^1(\Omega),$$

and we can obtain an analogous representation. We conclude that the classic Green's representation holds.

THEOREM 3.1. *For samples* $\{G^n, n = 1, \ldots, \mathcal{N}\}$, *we have*

$$(U^n, \psi) = (G^n, \phi) \tag{3.3}$$

for $n = 1, \ldots, \mathcal{N}$. *We also have*

$$E\big((U, \psi)\big) = (E(G), \phi). \tag{3.4}$$

The point is that, theoretically, instead of solving a partial differential equation for each sample in order to build the distribution of (U^n, ψ), we can solve one deterministic problem to get ϕ and then calculate values of (U^n, ψ) using a relatively inexpensive inner product. Indeed, we never approximate U^n in order to estimate the samples of the quantity of interest in this approach.

In order to make this approach practical, we introduce a finite element approximation $\phi_h \in \mathcal{V}_h$ satisfying

$$a^*(\phi_h, v) = (\psi, v), \quad v \in \mathcal{V}_h. \tag{3.5}$$

We obtain the computable approximations

$$(U^n, \psi) \approx (\phi_h, G^n), \tag{3.6}$$

$$E\big((U, \psi)\big) \approx \big(E(G), \phi_h\big). \tag{3.7}$$

3.2. A posteriori error estimate for samples. We next present an a posteriori error analysis for the approximate value for each sample (U^n, ψ) and for $E\big((U,\psi)\big)$. For samples, the error is

$$(U^n,\psi) - (G^n,\phi_h) = (G^n,\phi) - (G^n,\phi_h) = (G^n,\phi_h - \phi).$$

For each sample, this is a quantity of interest for ϕ corresponding to G^n. To avoid confusion, we let $\Theta^n \in \mathcal{H}_0^1(\Omega)$ denote the forward adjoint solution solving

$$a(\Theta^n, v) = (G^n, v), \quad v \in \mathcal{H}_0^1(\Omega). \tag{3.8}$$

Note that because we treat a linear problem, $\Theta^n = U^n$. Similarly, we let $E(\Theta) \in \mathcal{H}_0^1(\Omega)$ solve

$$a(E(\Theta), v) = (E(G), v), \quad v \in \mathcal{H}_0^1(\Omega). \tag{3.9}$$

The standard analysis gives

$$\begin{aligned}(G^n,\phi_h - \phi) &= a(\Theta^n,\phi_h - \phi) = a^*(\phi_h - \phi,\Theta^n)\\ &= a^*(\phi_h,\Theta^n) - a^*(\phi,\Theta^n) = a^*(\phi_h,\Theta^n) - (\psi,\Theta^n)\\ &= a^*(\phi_h,(I-\pi_h)\Theta^n) - (\psi,(I-\pi_h)\Theta^n),\end{aligned}$$

where the last step follows from Galerkin orthogonality. We can argue similarly for $E\big((G^n,\phi_h - \phi)\big)$.

To use this representation, we need to solve the forward adjoint problem (3.8) in $\mathcal{V}_{\tilde{h}}$. Unfortunately, in the case of the error of the samples, this requires computing n approximate forward adjoint solutions Θ^n using a more expensive finite element computation. Another approach is simply to approximate (G^n,ϕ) by $(G^n,\phi_{\tilde{h}})$, where $\phi_{\tilde{h}}$ is a finite element approximation of ϕ in $\mathcal{V}_{\tilde{h}}$. An analysis of the accuracy of this replacement follows easily from the relation

$$(G^n,\phi_h - \phi) = (G^n,\phi_h - \phi_{\tilde{h}}) + (G^n,\phi_{\tilde{h}} - \phi).$$

In either case, arguing as for Theorem 2.2 yields the following.

THEOREM 3.2. *The solution error in each sample is estimated by*

$$(G^n,\phi_h - \phi) \approx a^*(\phi_h,(I-\pi_h)\Theta^n) - (\psi,(I-\pi_h)\Theta^n), \tag{3.10}$$

where Θ^n is a finite element approximation for the adjoint problem (3.8) in $\mathcal{V}_{\tilde{h}}$ and π_h denotes a projection into the finite element space $\mathcal{V}_h$. We also have

$$(G^n,\phi_h - \phi) \approx (G^n,\phi_h - \phi_{\tilde{h}}), \tag{3.11}$$

where $\phi_{\tilde{h}}$ is a finite element solution of the adjoint problem (3.2) computed in $\mathcal{V}_{\tilde{h}}$.

We also have the estimates

$$E\big((G^n,\phi_h - \phi)\big) \approx a^*(\phi_h,(I-\pi_h)E(\Theta)) - (\psi,(I-\pi_h)E(\Theta)), \tag{3.12}$$

where $E(\Theta)$ is a finite element approximation for the adjoint problem (3.9) computed on a finer mesh $\mathcal{T}_{\tilde{h}}$ or using $\mathcal{V}_h^2$. We also have

$$E\big((G,\phi_h - \phi)\big) \approx (E(G),\phi_h - \tilde{\phi}_h). \tag{3.13}$$

The error of these estimates is bounded by Ch^3 or $C\tilde{h}^2$. In both cases, these bounds are asymptotically smaller than the estimates themselves.

4. A posteriori error analysis for an approximate distribution. We now present an a posteriori error analysis for the approximate cumulative distribution function obtained from $\mathcal{N}$ approximate sample values of a linear functional of a solution of a partial differential equation with random perturbations.

We let $U = U(X)$ be a solution of an elliptic problem that is randomly perturbed by a random variable X on a probability space $(\Omega, \mathcal{B}, P)$, and $Q(U) = (U, \psi)$ be a quantity of interest for some $\psi \in L^2(\Omega)$. We want to approximate the probability distribution function of $Q = Q(X)$,

$$F(t) = P\big(\{X : Q(U(X)) \le t\}\big) = P\big(Q \le t\big).$$

We use the sample distribution function computed from a finite collection of approximate sample values $\{\tilde{Q}^n, n = 1, \ldots, \mathcal{N}\} = \{(\tilde{U}_n, \psi), n = 1, \ldots, \mathcal{N}\}$:

$$\tilde{F}_{\mathcal{N}}(t) = \frac{1}{\mathcal{N}} \sum_{n=1}^{\mathcal{N}} I\big(\tilde{Q}^n \le t\big),$$

where I is the indicator function. Here, $\tilde{U}^n$ is a numerical approximation for a true solution U^n corresponding to a sample X^n. We assume that there is an error estimate

$$\tilde{Q}^n - Q^n \approx \mathcal{E}^n,$$

with $Q^n = (U^n, \psi)$. We use Theorem 2.2 or 3.2, for example.

There are two sources of error:

1. finite sampling,
2. numerical approximation of the differential equation solutions.

We define the sample distribution function

$$F_{\mathcal{N}}(t) = \frac{1}{\mathcal{N}} \sum_{n=1}^{\mathcal{N}} I\big(Q^n \le t\big)$$

and decompose the error

$$\big|F(t) - \tilde{F}_{\mathcal{N}}(t)\big| \le \big|F(t) - F_{\mathcal{N}}(t)\big| + \big|F_{\mathcal{N}}(t) - \tilde{F}_{\mathcal{N}}(t)\big| = I + II. \tag{4.1}$$

There is extensive statistics literature treating I; e.g., see [10]. We note that $F_{\mathcal{N}}$ has very desirable properties; e.g., see the following.

- As a function of t, $F_{\mathcal{N}}(t)$ is a distribution function, and for each fixed t, $F_{\mathcal{N}}(t)$ is a random variable corresponding to the sample.
- It is an unbiased estimator, i.e., $E(F_{\mathcal{N}}) \equiv E(F)$.
- $\mathcal{N}F_{\mathcal{N}}(t)$ has exact binomial distribution for $\mathcal{N}$ trials and probability of success $F(t)$.
- $\mathrm{Var}(F_{\mathcal{N}}(t)) = F(t)(1 - F(t))/\mathcal{N} \to 0$ as $\mathcal{N} \to \infty$, and $F_{\mathcal{N}}$ converges in mean square to F as $\mathcal{N} \to \infty$.

The approximation properties of $F_{\mathcal{N}}(t)$ can be studied in various ways, all of which have the flavor of bounding the error with high probability in the limit of large $\mathcal{N}$. One useful measure is the Kolmogorov–Smirnov distance

$$\sup_{t \in \mathbb{R}} \big|F_{\mathcal{N}}(t) - F(t)\big|.$$

A result that is useful for being uniform in t is that there is a constant $C > 0$ such that

$$P\left(\sup_{t\in\mathbb{R}} |F_{\mathcal{N}}(t) - F(t)| > \epsilon\right) \leq Ce^{-2\epsilon^2\mathcal{N}} \quad \text{for all } \epsilon > 0;$$

see [10]. We rewrite this as for any $\epsilon > 0$,

$$\sup_{t\in\mathbb{R}} |F_{\mathcal{N}}(t) - F(t)| \leq \left(\frac{\log(\epsilon^{-1})}{2\mathcal{N}}\right)^{1/2} \tag{4.2}$$

with probability greater than $1 - \epsilon$.

Another standard measure is the mean square error (MSE),

$$MSE(\tilde{\Theta}) = E\big((\tilde{\Theta} - \Theta)^2\big),$$

where $\tilde{\Theta}$ is an estimator for Θ. We define

$$\mathcal{X}_n(t) = \begin{cases} 1, & Q_n < t, \\ 0, & \text{otherwise}, \end{cases} \qquad \mathcal{X}(t) = \begin{cases} 1, & Q < t, \\ 0, & \text{otherwise}. \end{cases}$$

We have

$$F_{\mathcal{N}}(t) = \frac{1}{\mathcal{N}}\sum_{n=1}^{\mathcal{N}} \mathcal{X}_n(t), \qquad F(t) = E(\mathcal{X})(t).$$

For all t,

$$MSE\left(\frac{1}{\mathcal{N}}\sum_{n=1}^{\mathcal{N}} \mathcal{X}_n(t)\right) = \frac{\sigma^2}{\mathcal{N}}, \quad \sigma^2 = F(t)(1 - F(t)). \tag{4.3}$$

We can also estimate the (unknown) variance by defining

$$S_{\mathcal{N}}^2 = \frac{1}{\mathcal{N}}\sum_{n=1}^{\mathcal{N}} \left(\mathcal{X}_n(t) - \frac{1}{\mathcal{N}}\sum_{n=1}^{\mathcal{N}} \mathcal{X}_n(t)\right)^2.$$

$S_{\mathcal{N}}^2$ is a computable estimator for σ^2 and

$$MSE(S_{\mathcal{N}}^2) = \frac{2\mathcal{N}+1}{\mathcal{N}^2}\sigma^4. \tag{4.4}$$

Another useful result follows from the observation that $\{\mathcal{X}_n\}$ are independently and identically distributed Bernoulli variables. The Chebyshev inequality implies that for $\epsilon > 0$,

$$P\left(\left|E(\mathcal{X})(t) - \frac{1}{\mathcal{N}}\sum_{n=1}^{\mathcal{N}} \mathcal{X}_n(t)\right| \leq \left(\frac{F(t)(1 - F(t))}{\mathcal{N}\epsilon}\right)^{1/2}\right) > 1 - \epsilon, \quad \epsilon > 0. \tag{4.5}$$

To obtain a computable estimate, we note that

$$F(t)(1 - F(t)) = F_{\mathcal{N}}(t)(1 - F_{\mathcal{N}}(t)) + (F(t) - F_{\mathcal{N}}(t))(1 - F(t) - F_{\mathcal{N}}(t)).$$

Therefore using (4.5) along with the fact that $0 \le F$ and $F_{\mathcal{N}} \le 1$,

$$\left(\frac{F(t)(1-F(t))}{\mathcal{N}\epsilon}\right)^{1/2} \le \left(\frac{F_{\mathcal{N}}(t)(1-F_{\mathcal{N}}(t))}{\mathcal{N}\epsilon}\right)^{1/2} + \left(\frac{|F(t)-F_{\mathcal{N}}(t)|\,|1-F(t)-F_{\mathcal{N}}(t)|}{\mathcal{N}\epsilon}\right)^{1/2}$$
$$\le \left(\frac{F_{\mathcal{N}}(t)(1-F_{\mathcal{N}}(t))}{\mathcal{N}\epsilon}\right)^{1/2} + \frac{1}{2\mathcal{N}\epsilon}.$$

We conclude

(4.6)
$$P\left(\left|E(\mathcal{X})(t) - \frac{1}{\mathcal{N}}\sum_{n=1}^{\mathcal{N}} \mathcal{X}_n(t)\right| \le \left(\frac{F_{\mathcal{N}}(t)(1-F_{\mathcal{N}}(t))}{\mathcal{N}\epsilon}\right)^{1/2} + \frac{1}{2\mathcal{N}\epsilon}\right) > 1-\epsilon, \quad \epsilon > 0.$$

Next, we consider II in (4.1).

$$II = \left|\frac{1}{\mathcal{N}}\sum_{n=1}^{\mathcal{N}}\left(I(\tilde{Q}^n \le t) - I(Q \le t)\right)\right| = \left|\frac{1}{\mathcal{N}}\sum_{n=1}^{\mathcal{N}}\left(I(Q^n + \mathcal{E}^n \le t) - I(Q \le t)\right)\right|$$
$$= \left|\frac{1}{\mathcal{N}}\sum_{\substack{n=1\\ \mathcal{E}^n \le 0}}^{\mathcal{N}}\left(I(Q^n \le t \le Q^n + |\mathcal{E}^n|)\right) + \frac{1}{\mathcal{N}}\sum_{\substack{n=1\\ \mathcal{E}^n > 0}}^{\mathcal{N}}\left(I(Q^n - |\mathcal{E}^n| \le t \le Q^n)\right)\right|.$$

We estimate

$$II \le \left|\frac{1}{\mathcal{N}}\sum_{n=1}^{\mathcal{N}}\left(I(Q^n - |\mathcal{E}^n| \le t \le Q^n + |\mathcal{E}^n|)\right)\right|. \tag{4.7}$$

If instead we expand using $Q^n = \tilde{Q}^n - \mathcal{E}^n$, we obtain the computable estimate

$$|F_{\mathcal{N}}(t) - \tilde{F}_{\mathcal{N}}(t)| \le \left|\frac{1}{\mathcal{N}}\sum_{n=1}^{\mathcal{N}}\left(I(\tilde{Q}^n - |\mathcal{E}^n| \le t \le \tilde{Q}^n + |\mathcal{E}^n|)\right)\right|. \tag{4.8}$$

Setting $\mathcal{E} = \max \mathcal{E}^n$ in (4.7), we obtain

$$|F_{\mathcal{N}}(t) - \tilde{F}_{\mathcal{N}}(t)| \le |F_{\mathcal{N}}(t+\mathcal{E}) - F_{\mathcal{N}}(t-\mathcal{E})|. \tag{4.9}$$

Now

$$|F_{\mathcal{N}}(t+\mathcal{E}) - F_{\mathcal{N}}(t-\mathcal{E})| \le |F(t+\mathcal{E}) - F(t-\mathcal{E})| + |F(t+\mathcal{E}) - F_{\mathcal{N}}(t+\mathcal{E})| + |F(t-\mathcal{E}) - F_{\mathcal{N}}(t-\mathcal{E})|.$$

Using (4.2), for any $c > 0$,

$$|F(t \pm \mathcal{E}) - F_{\mathcal{N}}(t \pm \mathcal{E})| \le \left(\frac{\log(\epsilon^{-1})}{2\mathcal{N}}\right)^{1/2}$$

with probability greater than $1-\epsilon$. Therefore, for any $\epsilon > 0$,

$$|F_{\mathcal{N}}(t) - \tilde{F}_{\mathcal{N}}(t)| \le |F(t+\mathcal{E}) - F(t-\mathcal{E})| + 2\left(\frac{\log(\epsilon^{-1})}{2\mathcal{N}}\right)^{1/2} \tag{4.10}$$

with probability greater than $1-\epsilon$.

Note that

$$F_{\mathcal{N}}(t)(1-F_{\mathcal{N}}(t)) = \tilde{F}_{\mathcal{N}}(t)(1-\tilde{F}_{\mathcal{N}}(t)) + (F_{\mathcal{N}}(t)-\tilde{F}_{\mathcal{N}}(t))(1-F_{\mathcal{N}}(t)-\tilde{F}_{\mathcal{N}}(t)).$$

We can bound the second expression on the right-hand side using (4.9) or (4.10) to obtain a computable estimator for the variance of F.

We summarize the most useful results.

THEOREM 4.1. *For any* $\epsilon > 0$,

$$|F(t)-\tilde{F}_{\mathcal{N}}(t)| \leq \left(\frac{\tilde{F}_{\mathcal{N}}(t)(1-\tilde{F}_{\mathcal{N}}(t))}{\mathcal{N}\epsilon}\right)^{1/2} + 2\left|\frac{1}{\mathcal{N}}\sum_{n=1}^{\mathcal{N}}\left(I(\tilde{Q}^n - |\mathcal{E}^n| \leq t \leq \tilde{Q}^n + |\mathcal{E}^n|)\right)\right| + \frac{1}{2\mathcal{N}\epsilon} \tag{4.11}$$

with probability greater than $1-\epsilon$.

With L *denoting the Lipschitz constant of* F, *for any* $\epsilon > 0$,

$$|F(t)-\tilde{F}_{\mathcal{N}}(t)| \leq \left(\frac{F(t)(1-F(t))}{\mathcal{N}\epsilon}\right)^{1/2} + L\max_{1\leq n\leq \mathcal{N}}\mathcal{E}^n + 2\left(\frac{\log(\epsilon^{-1})}{2\mathcal{N}}\right)^{1/2} \tag{4.12}$$

with probability greater than $1-\epsilon$.

Remark 4.1. The leading order bounding terms in the a posteriori bound (4.11) are computable, while the remainder tends to zero more rapidly in the limit of large $\mathcal{N}$. The bound (4.12) is useful for the design of adaptive algorithms among other things. Assuming that the solutions of the elliptic problems are in $\mathcal{H}^2$, it indicates that the error in the computed distribution function is bounded by an expression in which the leading order is proportional to

$$\frac{1}{\sqrt{\epsilon\mathcal{N}}} + Lh^2$$

with probability $1-\epsilon$. This suggests that, in order to balance the error arising from finite sampling against the error in each computed sample, we typically should choose

$$\mathcal{N} \sim h^{-4}.$$

This presents a compelling argument for seeking efficient ways to compute samples and control the accuracy.

Remark 4.2. The expression

$$\left|\frac{1}{\mathcal{N}}\sum_{n=1}^{\mathcal{N}}\left(I(\tilde{Q}^n - |\mathcal{E}^n| \leq t \leq \tilde{Q}^n + |\mathcal{E}^n|)\right)\right|$$

is itself an expected value. If $\mathcal{M} < \mathcal{N}$ and

$$\mathcal{N}' = \{n_1 < \cdots < n_{\mathcal{M}}\}$$

is a set of integers chosen at random from $\{1,\ldots,\mathcal{N}\}$, we can use the unbiased estimator

$$\left|\frac{1}{\mathcal{M}}\sum_{n\in\mathcal{N}'}\left(I(\tilde{Q}^n - |\mathcal{E}^n| \leq t \leq \tilde{Q}^n + |\mathcal{E}^n|)\right)\right|, \tag{4.13}$$

which has error that decreases as $\mathbf{O}(1/\sqrt{\mathcal{M}})$. This is reasonable when $\mathcal{N}$ is large since we are likely to require less accuracy in the error estimate than in the primary quantity of interest.

Remark 4.3. A similar error analysis can be carried out for an arbitrary stochastic moment q with an unbiased estimator Q using $\mathcal{N}$ samples. We let $\tilde{X}$ be an approximation to X and decompose the error as

$$|q(X) - Q(\tilde{X})| \leq |q(X) - Q(X)| + |Q(X) - Q(\tilde{X})|.$$

The first term can be estimated using the Chebyshev inequality, for $\epsilon > 0$,

$$P\left(|q(X) - Q(X)| \leq \left(\frac{\mathrm{Var}(Q(X))}{\epsilon \mathcal{N}}\right)^{1/2}\right) > 1 - \epsilon.$$

Since the variance of $Q(X)$ decreases as $\mathcal{N}$ increases, we obtain estimates for this term analogous to the expressions above. We can estimate the numerical error $Q(X) - Q(\tilde{X})$ by computing a solution on a finer mesh $Q(\hat{X}) - Q(\tilde{X})$. In the particular case that $q(X) = E[X]$, we can compute the quantity very efficiently; see section 3.

4.1. A numerical example. We illustrate the accuracy of the computable bound (4.11) using some simple experiments. We emphasize that (4.11) is a bound, and, in particular, we trade accuracy in terms of estimating the size of the error by increasing the probability that the bound is larger than the error. In this case, we desire that the degree of overestimation does not depend strongly on the discretization parameters.

To carry out the test, we specify a true cumulative distribution function (c.d.f.) and sample $\mathcal{N}$ points at random from the distribution. To each sample value, we add a random error drawn at random from another distribution. We use the Kaplar–Meier estimate for the approximate c.d.f. in the Matlab statistics toolbox and then compute the difference with the true c.d.f. values at the sample points. We also compute the difference divided by the true c.d.f. values.

The experiments we report include the following.

First computation	
Sample distribution	Normal, mean 1, variance 2
Error distribution	Uniform on $[-\delta, \delta]$
Second computation	
Sample distribution	Exponential, parameter 1
Error distribution	Uniform on $[-\delta, \delta]$
Third computation	
Sample distribution	Exponential, parameter 1
Error distribution	Uniform on $[-\delta X, \delta X]$, X = sample value

We obtained similar results for a variety of examples.

In Figure 4.1, we present three examples of approximate c.d.f. functions. In all cases, we bound the error with probability greater than 95%. In Figure 4.2, we plot the 95% confidence level bound calculated from (4.11) and compare this to the actual errors.

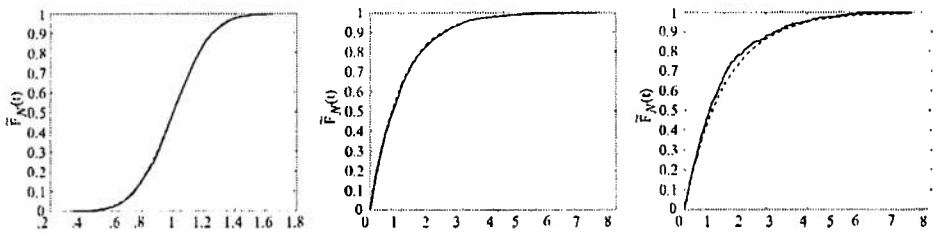


FIG. 4.1. *Plots of approximate (solid line) and true (dashed line) c.d.f. functions. Left: first computation with* $\mathcal{N} = 5000$, $\delta = .001$. *Middle: second computation with* $\mathcal{N} = 2000$, $\delta = .0001$. *Right: third computation with* $\mathcal{N} = 500$, $\delta = .05$.

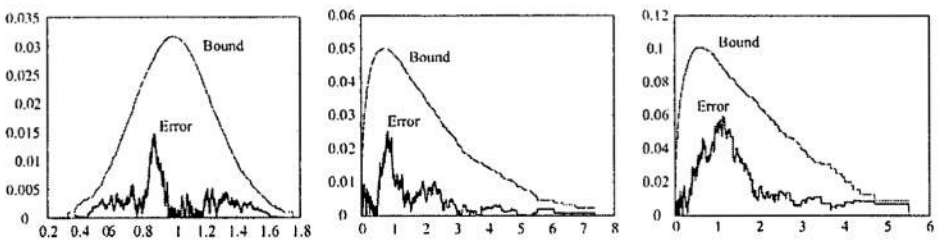


FIG. 4.2. *Plots of the 95% confidence level bound calculated from* (4.11) *for the examples shown in Figure* 4.1. *Left: first computation with* $\mathcal{N} = 5000$, $\delta = .001$. *Middle: second computation with* $\mathcal{N} = 2000$, $\delta = .0001$. *Right: third computation with* $\mathcal{N} = 500$, $\delta = .05$.

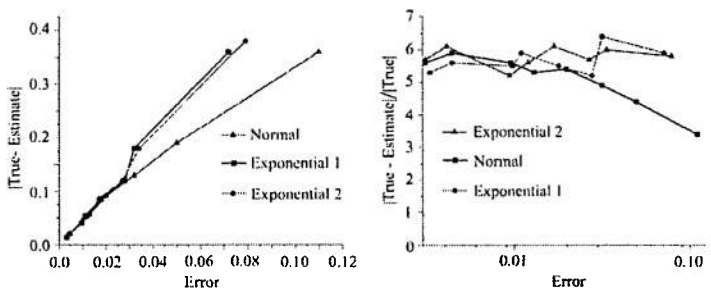


FIG. 4.3. *Performance of the bound* (4.11) *for the three examples shown in Figure* 4.1. *Left: plot of the difference between the estimate and bound versus the error. Right: plot of the relative difference versus the error.*

In Figure 4.3, we plot the performance of the bound with respect to estimating the size of the error. In all three cases, the bound is asymptotically around 5 times too large.

5. Adaptive error control. We now use Theorems 2.2, 3.2, and 4.1 to construct an adaptive error control algorithm. The computational parameters we wish to optimize are the mesh size h, the number of terms in the truncated Neumann series $\mathcal{P}$, the number of iterations in the domain decomposition algorithm $\mathcal{I}$, and the number of samples $\mathcal{N}$. The first task is to express the error $\mathcal{E}$ as a sum of three terms corresponding, respectively, to discretization error, error from the incomplete Neumann series, and error from the incomplete domain decomposition iteration.

Considering the problem with randomly perturbed diffusion coefficient, we bound the leading expression in the error estimate (2.9) in one sample value as

$$\begin{aligned}\mathcal{E}^n &= \left|(f,\Phi^n_{\mathcal{P},\mathcal{I}}) - (\mathbb{A}^n\nabla U^n_{\mathcal{P},\mathcal{I}},\nabla\Phi^n_{\mathcal{P},\mathcal{I}})\right| \\ &\le \left|(f,\Phi^n_{\mathcal{P},\mathcal{I}}) - (\mathbb{A}^n\nabla U^n_{\mathcal{P},\mathcal{I}},\nabla\Phi^n_{\mathcal{P},\mathcal{I}}) - (\mathbb{A}^n\nabla(U^n_{\infty,\infty}-U^n_{\mathcal{P},\mathcal{I}}),\nabla\Phi^n_{\mathcal{P},\mathcal{I}})\right| \\ &\quad + \left|(\mathbb{A}^n\nabla(U^n_{\mathcal{P},\infty}-U^n_{\mathcal{P},\mathcal{I}}),\nabla\Phi^n_{\mathcal{P},\mathcal{I}})\right| \\ &\quad + \left|(\mathbb{A}^n\nabla(U^n_{\infty,\infty}-U^n_{\mathcal{P},\infty}),\nabla\Phi^n_{\mathcal{P},\mathcal{I}})\right| \\ &= \mathcal{E}^n_{\mathrm{I}} + \mathcal{E}^n_{\mathrm{II}} + \mathcal{E}^n_{\mathrm{III}},\end{aligned} \tag{5.1}$$

where we use the obvious notation to denote the quantities obtained by taking $\mathcal{I},\mathcal{P}\to\infty$.

The goal is to estimate $\mathcal{E}^n_{\mathrm{I}},\mathcal{E}^n_{\mathrm{II}}$, and $\mathcal{E}^n_{\mathrm{III}}$ in terms of computable quantities. To do this, we introduce $\Delta\mathcal{I}$ and $\Delta\mathcal{P}$ as positive integers and use the approximations

$$U^n_{\mathcal{P}+\Delta\mathcal{P},\mathcal{I}+\Delta\mathcal{I}} \approx U^n_{\infty,\infty}.$$

The accuracy of the estimates below improves as $\Delta\mathcal{I}$ and $\Delta\mathcal{P}$ increase.

We have

$$\mathcal{E}^n_{\mathrm{I}} \approx \left|(f,\Phi^n_{\mathcal{P},\mathcal{I}}) - (\mathbb{A}^n\nabla U^n_{\mathcal{P},\mathcal{I}},\nabla\Phi^n_{\mathcal{P},\mathcal{I}}) - (\mathbb{A}^n\nabla(U^n_{\mathcal{P}+\Delta\mathcal{P},\mathcal{I}+\Delta\mathcal{I}}-U^n_{\mathcal{P},\mathcal{I}}),\nabla\Phi^n_{\mathcal{P},\mathcal{I}})\right|. \tag{5.2}$$

Likewise, we estimate

$$\mathcal{E}^n_{\mathrm{II}} \approx \left|(\mathbb{A}^n\nabla(U^n_{\mathcal{P},\mathcal{I}+\Delta I}-U^n_{\mathcal{P},\mathcal{I}}),\nabla\Phi^n_{\mathcal{P},\mathcal{I}})\right|, \tag{5.3}$$

$$\mathcal{E}^n_{\mathrm{III}} \approx \left|(\mathbb{A}^n\nabla(U^n_{\mathcal{P}+\Delta\mathcal{P},\mathcal{I}}-U^n_{h,\mathcal{P},\mathcal{I}}),\nabla\Phi^n_{\mathcal{P},\mathcal{I}})\right|. \tag{5.4}$$

We can find other expressions for $\mathcal{E}^n_{\mathrm{III}}$ by passing to the limit in (2.3) on each domain d to write

$$\vec{U}^{n,d}_{\infty,\infty} = \sum_{p=0}^{\infty}\left((-A^{n,d})^p((\mathbf{k}^{a,d})^{-1}\mathbf{k}^d)^p\right)\,(\mathbf{k}^{a,d})^{-1}\left(\vec{b}^d(f)+\vec{b}^{n,d}(\mathbb{A}^n,U^{n,d'}_{\infty,\infty})\right),$$

while

$$\vec{U}^{n,d}_{\mathcal{P},\infty} = \sum_{p=0}^{\mathcal{P}-1}\left((-A^{n,d})^p((\mathbf{k}^{a,d})^{-1}\mathbf{k}^d)^p\right)\,(\mathbf{k}^{a,d})^{-1}\left(\vec{b}^d(f)+\vec{b}^{n,d}(\mathbb{A}^n,U^{n,d'}_{\mathcal{P},\infty})\right).$$

Subtracting and approximating, we find

$$\begin{aligned}&\vec{U}^{n,d}_{\infty,\infty} - \vec{U}^{n,d}_{\mathcal{P},\infty} \\ &\quad\approx \sum_{p=\mathcal{P}}^{\infty}\left((-A^{n,d})^p((\mathbf{k}^{a,d})^{-1}\mathbf{k}^d)^p\right)\,(\mathbf{k}^{a,d})^{-1}\left(\vec{b}^d(f)+\vec{b}^{n,d}(\mathbb{A}^n,U^{n,d'}_{\infty,\infty})\right) \\ &\qquad+\sum_{p=0}^{\mathcal{P}-1}\left((-A^{n,d})^p((\mathbf{k}^{a,d})^{-1}\mathbf{k}^d)^p\right)\,(\mathbf{k}^{a,d})^{-1}\left(\vec{b}^{n,d}(\mathbb{A}^n,U^{n,d'}_{\infty,\infty})-\vec{b}^{n,d}(\mathbb{A}^n,U^{n,d'}_{\mathcal{P},\infty})\right).\end{aligned}$$

Summing yields

$$\begin{aligned}
&\vec{U}^{n,d}_{\infty,\infty} - \vec{U}^{n,d}_{\mathcal{P},\infty} \\
&= \left((-A^{n,d})^{\mathcal{P}}((\mathbf{k}^{a,d})^{-1}\mathbf{k}^{d})^{\mathcal{P}}\right)\, \vec{U}^{n,d}_{\infty,\infty} \\
&\quad + \sum_{p=0}^{\mathcal{P}-1} \left((-A^{n,d})^{p}((\mathbf{k}^{a,d})^{-1}\mathbf{k}^{d})^{p}\right)\, (\mathbf{k}^{a,d})^{-1}\left(\vec{b}^{n,d}(\mathbb{A}^{n}, U^{n,d'}_{\infty,\infty}) - \vec{b}^{n,d}(\mathbb{A}^{n}, U^{n,d'}_{\mathcal{P},\infty})\right).
\end{aligned}$$

Finally, approximating yields

$$\begin{aligned}
&\vec{U}^{n,d}_{\infty,\infty} - \vec{U}^{n,d}_{\mathcal{P},\infty} \\
&\approx \left((-A^{n,d})^{\mathcal{P}}((\mathbf{k}^{a,d})^{-1}\mathbf{k}^{d})^{\mathcal{P}}\right)\, \vec{U}^{n,d}_{\mathcal{P},\mathcal{I}} \\
&\quad + \sum_{p=0}^{\mathcal{P}-1} \left((-A^{n,d})^{p}((\mathbf{k}^{a,d})^{-1}\mathbf{k}^{d})^{p}\right)\, (\mathbf{k}^{a,d})^{-1}\left(\vec{b}^{n,d}(\mathbb{A}^{n}, U^{n,d'}_{\mathcal{P}+\Delta\mathcal{P},\mathcal{I}}) - \vec{b}^{n,d}(\mathbb{A}^{n}, U^{n,d'}_{\mathcal{P},\mathcal{I}})\right).
\end{aligned}$$

We denote the operators corresponding to $\mathbf{k}^a$ and $\mathbf{k}$ on $\mathcal{V}_{h,d}$ by k^a and k, respectively. We have

$$\begin{aligned}
(5.5)\quad \mathcal{E}^{n}_{\mathrm{III}} \approx \sum_{d=1}^{\mathcal{D}} \Bigg(&\mathbb{A}^{n}\nabla((-A^{n,d})^{\mathcal{P}}((k^{a,d})^{-1}k^{d})^{\mathcal{P}})U^{n,d}_{\mathcal{P},\mathcal{I}}, \nabla\Phi^{n}_{\tilde{\mathcal{P}},\tilde{\mathcal{I}}}\big)_{d} \\
+ \sum_{p=0}^{\mathcal{P}-1} &\left((-A^{n,d})^{p}((\mathbf{k}^{a,d})^{-1}\mathbf{k}^{d})^{p}\right)\, (\mathbf{k}^{a,d})^{-1}\left(\vec{b}^{n,d}(\mathbb{A}^{n}, U^{n,d'}_{\mathcal{P}+\Delta\mathcal{P},\mathcal{I}}) - \vec{b}^{n,d}(\mathbb{A}^{n}, U^{n,d'}_{\mathcal{P},\mathcal{I}})\right)\Bigg).
\end{aligned}$$

We now present an adaptive error control strategy in Algorithm 3 based on Theorem 4.1 and the approximations

$$\mathcal{E}^{n} \approx \hat{\mathcal{E}}^{n} = \mathcal{E}^{n}_{\mathrm{I}} + \mathcal{E}^{n}_{\mathrm{II}} + \mathcal{E}^{n}_{\mathrm{III}}.$$

We set

$$\mathcal{E}_{\mathrm{I}} = \max_{n} \mathcal{E}^{n}_{\mathrm{I}},\ \ \mathcal{E}_{\mathrm{II}} = \max_{n} \mathcal{E}^{n}_{\mathrm{II}},\ \ \mathcal{E}_{\mathrm{III}} = \max_{n} \mathcal{E}^{n}_{\mathrm{III}}.$$

We define in addition

$$(5.6)\qquad \mathcal{E}_{\mathrm{IV}} = \left(\frac{\tilde{F}_{\mathcal{N}}(t)(1-\tilde{F}_{\mathcal{N}}(t))}{\mathcal{N}\epsilon}\right)^{1/2}$$

for a given $\epsilon > 0$.

5.1. A numerical example. We apply the adaptive algorithm to the problem given in section 2.4. We start with a coarse mesh and small number of iterations, terms, and samples and let the adaptive algorithm choose the parameter values in order to make the error bound of $F(t)$ smaller than 15% with 95% likelihood; i.e., we set TOL = 0.15 and $\epsilon = .05$. We set $\sigma_1 = 0.5$, $\sigma_2 = \sigma_3 = 0.125$, and $\sigma_4 = 0.25$.

Initially, we let $h = 1/18$ determine a uniform initial mesh, $\mathcal{I} = 40$, $\mathcal{P} = 1$, and $\mathcal{N} = 60$. We set $\Delta\mathcal{I} = 0.3\mathcal{I}$ and $\Delta\mathcal{P} = 1$. We compute the adjoint solution using a refined mesh with $\tilde{h} = h/2$ but using the same number of iterations, terms, and samples as the forward problem. To refine, we set $h_i = 1/(9(i-1))$, with $i = 3$

Algorithm 3. ADAPTIVE ALGORITHM

Choose ϵ in Theorem 4.1, which determines the reliability of the error control
Let TOL be the desired tolerance of the error $|F(t) - \bar{F}_{\mathcal{N}}(t)|$
Let $\sigma_1+\sigma_2+\sigma_3+\sigma_4 = 1$ be positive numbers that are used to apportion the tolerance TOL between the four contributions to the error, with values chosen based on the computational cost associated with changing the four discretization parameters
Choose initial meshes $\mathcal{T}_h$, $\mathcal{T}_{\tilde{h}}$ and $\mathcal{P}$, $\mathcal{I}$, and $\mathcal{N}$

Compute $\{U^n_{\mathcal{P},\mathcal{I}}, n = 1,\ldots,\mathcal{N}\}$ in the space $\mathcal{V}_h$ and the sample quantity of interest values
Compute $\{\Phi^n_{\mathcal{P},\tilde{\mathcal{I}}}, n = 1,\ldots,\mathcal{N}\}$ in the space $\mathcal{V}_{\tilde{h}}$
Compute $\mathcal{E}^n_{\mathrm{I}}, \mathcal{E}^n_{\mathrm{II}}, \mathcal{E}^n_{\mathrm{III}}$ for $n = 1,\ldots,\mathcal{N}$

Compute $\bar{F}_{\mathcal{N}}(t)$
Estimate the Lipschitz constant L of F using $\bar{F}_{\mathcal{N}}$
Compute $\mathcal{E}_{\mathrm{IV}}$

while $\mathcal{E}_{\mathrm{IV}} + \left|\frac{1}{\mathcal{N}}\sum_{n=1}^{\mathcal{N}}\left(I(\tilde{Q}^n - |\hat{\mathcal{E}}^n| \le t \le \tilde{Q}^n + |\hat{\mathcal{E}}^n|)\right)\right| \ge \mathrm{TOL}$ **do**

if $L\mathcal{E}_{\mathrm{I}} > \sigma_1\mathrm{TOL}$ **then**
Refine $\mathcal{T}_h$ and $\mathcal{T}_{\tilde{h}}$ to meet the prediction that $\mathcal{E}_{\mathrm{I}} \approx \sigma_1\mathrm{TOL}$ on the new mesh
end if
if $L\mathcal{E}_{\mathrm{II}} > \sigma_2\mathrm{TOL}$ **then**
Increase $\mathcal{P}$ to meet the prediction $\mathcal{E}_{\mathrm{II}} \approx \sigma_2\mathrm{TOL}$
end if
if $L\mathcal{E}_{\mathrm{III}} > \sigma_3\mathrm{TOL}$ **then**
Increase $\mathcal{P}$ to meet the prediction $\mathcal{E}_{\mathrm{III}} \approx \sigma_3\mathrm{TOL}$
end if
if $\mathcal{E}_{\mathrm{IV}} > \sigma_4\mathrm{TOL}$ **then**
Increase $\mathcal{N}$ to meet the prediction $\mathcal{E}_{\mathrm{IV}} \approx \sigma_4\mathrm{TOL}$
end if

Compute $\{U^n_{\mathcal{P},\mathcal{I}}, n = 1,\ldots,\mathcal{N}\}$ in the space $\mathcal{V}_h$ and the sample quantity of interest values
Compute $\{\Phi^n_{\mathcal{P},\tilde{\mathcal{I}}}, n = 1,\ldots,\mathcal{N}\}$ in the space $\mathcal{V}_{\tilde{h}}$
Compute $\mathcal{E}^n_{\mathrm{I}}, \mathcal{E}^n_{\mathrm{II}}, \mathcal{E}^n_{\mathrm{III}}$ for $n = 1,\ldots,\mathcal{N}$

Compute $\bar{F}_{\mathcal{N}}(t)$
Estimate the Lipschitz constant L of F using $\bar{F}_{\mathcal{N}}$
Compute $\mathcal{E}_{\mathrm{IV}}$

end while

initially, and then for each refinement we increment i by 2. This means that we get 3, 5, 7, etc. nodes in the x-direction and y-direction on each subdomain.

In Figure 5.1 we present the parameter values for each of the iterates. The tolerance was reached after three iterations with $h = 1/54$, $\mathcal{I} = 160$, $\mathcal{P} = 3$, and $\mathcal{N} = 240$. In Figure 5.2, we plot error bound indicators after each iteration in the

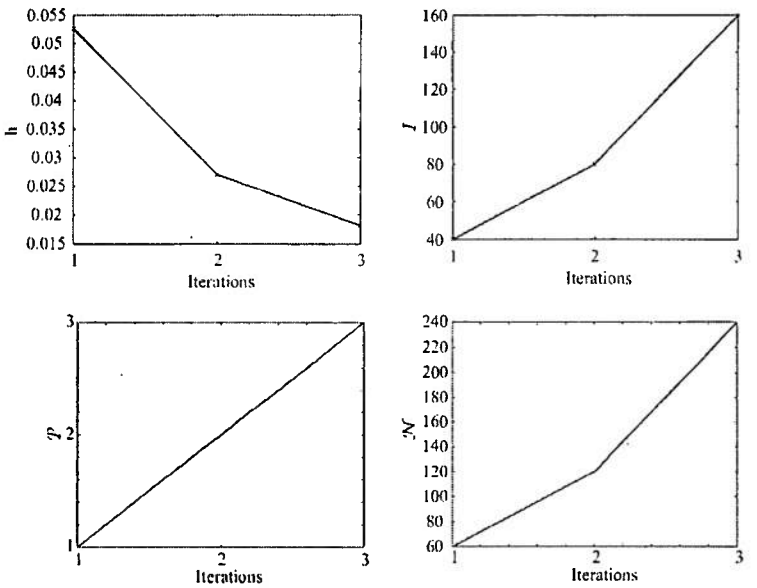


FIG. 5.1. *Computational parameters chosen adaptively according to the adaptive algorithm.*

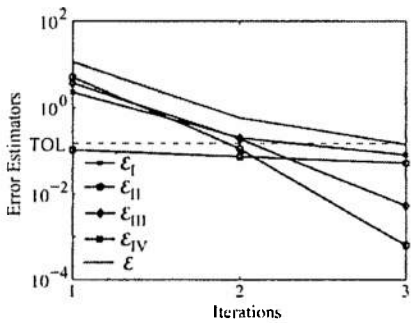


FIG. 5.2. *The error estimators computed after each iteration in the adaptive algorithm.*

adaptive algorithm and the total error bound. We compute an approximate error using a reference solution with $h = 1/72$, $\mathcal{I} = 300$, $\mathcal{P} = 5$, and $\mathcal{N} = 480$ and show the result in Figure 5.3. The error decreases from almost 100% initially, with a distribution function that fails to detect critical behavior, to an error of around 30% to finally an error less than 3%.

6. Conclusion. In this paper, we consider the nonparametric density estimation problem for a quantity of interest computed from solutions of an elliptic partial differential equation with randomly perturbed coefficients and data. We focused on problems for which limited knowledge of the random perturbations is known. In particular, we assume that the random perturbation to the diffusion coefficient is described by a piecewise constant function. We derive an efficient method for computing samples and generating an approximate probability distribution based on Lion's domain decomposition method and the Neumann series. We then derive an a posteriori

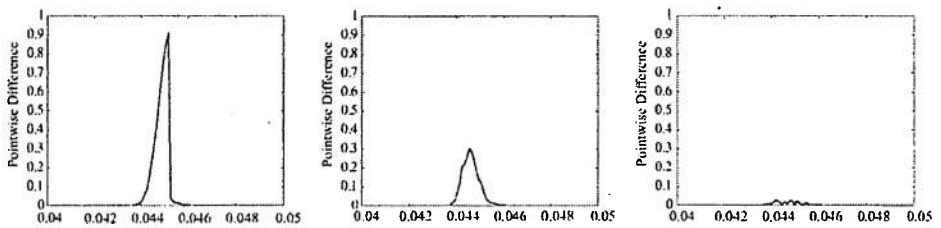


FIG. 5.3. *Approximate error in the solutions produced by the adaptive algorithm for iterations* 1, 2, *and* 3.

error estimate for the computed probability distribution reflecting all sources of deterministic and statistical errors, including discretization of the domain, finite iteration of the domain decomposition iteration, finite truncation in the Neumann series, and the effect of using a finite number of random samples. Finally, we develop an adaptive error control algorithm based on the a posteriori estimate.

REFERENCES

[1] I. BABUŠKA, R. TEMPONE, AND G. E. ZOURARIS, *Solving elliptic boundary value problems with uncertain coefficients by the finite element method: The stochastic formulation*, Comput. Methods Appl. Mech. Engrg., 194 (2005), pp. 1251–1294.

[2] W. BANGERTH AND R. RANNACHER, *Adaptive Finite Element Methods for Differential Equations*, Lectures Math. ETH Zürich, Birkhäuser Verlag, Basel, 2003.

[3] D. ESTEP, M. J. HOLST, AND A. MÅLQVIST, *Nonparametric density estimation for randomly perturbed elliptic problems III: Convergence and a priori analysis*, in preparation.

[4] D. ESTEP, M. G. LARSON, AND R. D. WILLIAMS, *Estimating the error of numerical solutions of systems of reaction-diffusion equations*, Mem. Amer. Math. Soc., 146 (2000), pp. 1–109.

[5] D. ESTEP, A. MÅLQVIST, AND S. TAVENER, *Nonparametric density estimation for randomly perturbed elliptic problems II: Applications and adaptive modeling*, Internat. J. Numer. Methods Engrg., to appear.

[6] C. L. FARMER, *Upscaling: A review*, Internat. J. Numer. Methods Fluids, 40 (2002), pp. 63–78.

[7] R. GHANEM AND P. SPANOS, *Stochastic Finite Elements: A Spectral Approach*, Dover, New York, 2003.

[8] W. GUO AND L. S. HOU, *Generalizations and accelerations of Lions' nonoverlapping domain decomposition method for linear elliptic PDE*, SIAM J. Numer. Anal., 41 (2003), pp. 2056–2080.

[9] P.-L. LIONS, *On the Schwarz alternating method. III. A variant for nonoverlapping subdomains*, in Third International Symposium on Domain Decomposition Methods for Partial Differential Equations (Houston, TX, 1989), SIAM, Philadelphia, PA, 1990, pp. 202–223.

[10] R. SERFLING, *Approximation Theorems of Mathematical Statistics*, John Wiley and Sons, New York, 1980.

SIAM J. NUMER. ANAL.
Vol. 49, No. 5, pp. 1836–1859

A MEASURE-THEORETIC COMPUTATIONAL METHOD FOR INVERSE SENSITIVITY PROBLEMS I: METHOD AND ANALYSIS*

J. BREIDT†, T. BUTLER‡, AND D. ESTEP†

Abstract. We consider the inverse sensitivity analysis problem of quantifying the uncertainty of inputs to a deterministic map given specified uncertainty in a linear functional of the output of the map. This is a version of the model calibration or parameter estimation problem for a deterministic map. We assume that the uncertainty in the quantity of interest is represented by a random variable with a given distribution, and we use the law of total probability to express the inverse problem for the corresponding probability measure on the input space. Assuming that the map from the input space to the quantity of interest is smooth, we solve the generally ill-posed inverse problem by using the implicit function theorem to derive a method for approximating the set-valued inverse that provides an approximate quotient space representation of the input space. We then derive an efficient computational approach to compute a measure theoretic approximation of the probability measure on the input space imparted by the approximate set-valued inverse that solves the inverse problem.

Key words. adjoint problem, density estimation, inverse sensitivity analysis, model calibration, nonparametric density estimation, parameter estimation, sensitivity analysis, set-valued inverse

AMS subject classifications. 60-08, 34F05

DOI. 10.1137/100785946

1. Introduction.

We develop and analyze a numerical method to solve the inverse sensitivity analysis problem: Given a specified variation and/or uncertainty in the output of a smooth map, determine variations in the input parameters that produce the observed uncertainty. We formulate this inverse problem using probability to describe variation by assuming that the inputs and outputs are random variables. This inverse problem has an abstract interpretation in which the density is imposed on the output in order to observe the consequences for the inputs. It also has an experimental interpretation in which the model output matches observed values of an experiment and the imposed density is associated with the experimental data, i.e., reflecting the uncertainty in the data or arising as a consequence of experimental error.

*Received by the editors February 16, 2010; accepted for publication (in revised form) June 24, 2011; published electronically September 20, 2011.
http://www.siam.org/journals/sinum/49-5/78594.html

†Department of Statistics, Colorado State University, Fort Collins, CO 80523 (jbreidt@stat.colostate.edu, estep@math.colostate.edu). The first author's work was supported in part by the National Aeronautics and Space Administration, Earth Sciences Division (#NNX08AK08G) the National Science Foundation (SES-0922142), and the Joint NSF/NIGMS Initiative to Support Research in the Area of Mathematical Biology (#R01GM096192). The content is solely the responsibility of the authors and does not necessarily represent the official views of the National Institute Of General Medical Sciences or the National Institutes of Health. The last author's work was supported in part by the Defense Threat Reduction Agency (HDTRA1-09-1-0036), Department of Energy (DE-FG02-04ER25620, DE-FG02-05ER25699, DE-FC02-07ER54909, DE-SC0001724), Lawrence Livermore National Laboratory (B573139, B584647), the National Aeronautics and Space Administration (NNG04GH63G), the National Science Foundation (DMS-0107832, DMS-0715135, DGE-0221595003, MSPA-CSE-0434354, ECCS-0700559), Idaho National Laboratory (00069249), NSF/NIGMS (#R01GM096192), and the Sandia Corporation (PO299784).

‡Institute for Computational Engineering and Sciences, University of Texas at Austin, Austin, TX 78712 (tbutler@ices.utexas.edu). This author's work was supported in part by the Department of Energy (DE-FG02-05ER25699) and the National Science Foundation (DGE-0221595003, MSPA-CSE-0434354).

To motivate this inverse sensitivity analysis problem, consider the situation of a manufacturer who will purchase a large number of metal plates of a given alloy and thickness that are to be used subsequently in a high temperature environment. In order to ensure the plates maintain integrity, the manufacturer specifies that a given heat load must be distributed quasi-uniformly after ten minutes of exposure, with some conditions on how much the temperature may vary through the plate. The plates are milled with variations in the purity of the alloy and the thickness of the plates, both of which affect the heat distribution under load. To check a batch of plates to see if it meets the requirements, the manufacturer tests the heat specification on a random sample of plates drawn from the batch. The random selection of samples, the variation in plate properties, and measurement error combined lead to a description of the test results as a random variable. After delivery, the manufacturer decides that knowing the statistics on the size of the plates and the composition of the alloy would be useful. The heat equation models the heat distribution under a given load once the conductivity determined by the alloy composition and the thickness of the plates are specified. The inverse sensitivity problem is to determine the distribution on the space of parameters consisting of the thickness and alloy purity from the distribution of the results of the heat experiments on the plates.

The probabilistic inverse problem can be described more precisely as follows.

Given

- a model $M(Y, \lambda)$ with solution $Y = G(\lambda)$ depending on parameters and data λ in parameter space $\mathbf{\Lambda} \subset \mathbb{R}^d$,
- a linear functional $q(\lambda) = q(Y(\lambda))$ taking values in an output space $\mathcal{D}$,
- an *observed* probability density $\rho_{\mathcal{D}}(q(\lambda)) = \rho_{\mathcal{D}}(q(Y(\lambda)))$ on the output value $q(\lambda)$,

determine

- a probability density $\sigma_{\mathbf{\Lambda}}(\lambda)$ on the parameter space $\mathbf{\Lambda}$ that produces the observed density.

We assume the model $M(Y, \lambda)$ depends smoothly on the inputs, so the map $q(\lambda)$ is implicitly a smooth and *deterministic* function of λ.

There are several important issues associated with this problem. In general, the parameter space is multidimensional while there is a single observation (or a low dimensional set of observations at most). So, the inverse problem is ill-posed in the sense that the inverse solution of the deterministic model is set-valued. Under the assumption of a smooth model, we address this issue by constructing a systematic method for approximating set-valued inverses. Second, we are particularly interested in models that are complicated and/or expensive to evaluate, e.g., requiring the solution of a differential equation, so that the map to the output is determined implicitly. We address this issue by using adjoint operators [22, 20, 6, 21, 23, 12, 13, 9, 10, 7, 11] to compute the required derivative information. Third, while probability densities describe random variables, the densities themselves are not random. Common approaches to approximating probability densities often use a random representation obtained by some variation of Monte Carlo sampling [14, 17, 18]; however, this is not a requirement. In particular, the approach described in this paper is not stochastic, rather it is based on the simple approximation commonly used in measure theory.

In this paper, we present the basic method and analysis of a measure-theoretic computational approach for the probabilistic inverse sensitivity analysis problem. In [4], we present a numerical analysis of the discretization error that arises when evaluating the model by numerical solution and using a finite number of random samples to represent the distribution on the output quantity. In [5], we discuss the problem

of dealing with multiple quantities of interest, which has application to data assimilation and "cascaded" uncertainty in operator decomposition solution of multiphysics problems.

This paper is structured as follows. In section 2, we formulate the probabilistic inverse problem that we solve and discuss the relation to a Bayesian inverse problem. In section 3.1, we deal with the set-valued nature of the inverse problem by introducing a theory of generalized contours and explain how the generalized contours can be approximated. In section 3.2, we develop a computational measure theoretic method for approximating the inverse parameter distribution using approximate generalized contours. In section 4, we apply the method to a variety of problems. Finally, section 5 summarizes the work.

2. Formulation of the probabilistic inverse problem. The inverse problem we study is the direct inversion of the forward stochastic sensitivity analysis problem for a deterministic model. We consider a deterministic operator $q(\lambda)$ that maps values in a parameter space Λ to an output space $\mathcal{D}$. We assume there is a *parameter volume* measure μ_Λ on Λ that determines the volume of sets in Λ. The volume measure depends on the units of measure used for the parameters and also reflects the structural dependency among the parameters, e.g., depending on whether or not μ_Λ is a product measure. The volume measure is specified as part of the model that defines the map $q(\lambda)$ since the parameters must be explicitly defined in the physical model that determines q. We assume that μ_Λ is absolutely continuous with respect to the Lebesgue measure and the volume V of Λ is finite.

We first describe the forward stochastic sensitivity analysis for the deterministic map $q(\lambda)$. We assume that a probability density $\sigma_\Lambda(\lambda)$ is specified on the parameter space Λ. This density distinguishes the probability of different events in Λ, i.e., the probability of an event A in Λ, by which we mean a measurable set of values, is computed via

$$P(A) = \int_A \sigma_\Lambda(\lambda)\, d\mu_\Lambda(\lambda).$$

The deterministic model can be expressed in terms of a likelihood function $L(q \mid \lambda)$ of the output q values given the input parameter values λ, where $L(q \mid \lambda) = \delta(q - q(\lambda))$ is the unit mass distribution at $q = q(\lambda)$. This implies the fundamental relationship

$$\text{(2.1)} \qquad \textit{Law of Total Probability} \qquad \rho_{\mathcal{D}}(q|A) = \frac{\int_A L(q \mid \lambda)\sigma_\Lambda(\lambda)\, d\mu_\Lambda(\lambda)}{\int_A \sigma_\Lambda(\lambda)\, d\mu_\Lambda(\lambda)}.$$

This is a Fredholm integral equation of the first kind that determines a conditional probability density $\rho_{\mathcal{D}}(q|A)$ on the output given that the parameters come from A. Thus, we may determine the conditional probability of event $B \subset \mathcal{D}$ as

$$P(B|A) = \int_B \rho_{\mathcal{D}}(q|A)\, d\mu_{\mathcal{D}}(q) = \frac{\int_B \int_A L(q \mid \lambda)\sigma_\Lambda(\lambda)\, d\mu_\Lambda(\lambda)\, d\mu_{\mathcal{D}}(q)}{\int_A \sigma_\Lambda(\lambda)\, d\mu_\Lambda(\lambda)}.$$

For forward sensitivity analysis it is common to take $A = \Lambda$ so that $P(B|A) = P(B)$, and we arrive at the common form for the law of total probability given by

$$\text{(2.2)} \qquad \rho_{\mathcal{D}}(q) = \int_\Lambda L(q \mid \lambda)\sigma_\Lambda(\lambda)\, d\mu_\Lambda(\lambda).$$

This describes an analogue of a Perron–Frobenius map where the deterministic map $q(\lambda)$ defines a transformation of the density $\sigma_\Lambda(\lambda)$ to $\rho_\mathcal{D}(q)$. This forward sensitivity analysis problem is often solved using a Monte Carlo approach: Random parameter sample values λ are drawn from the distribution σ_Λ on the parameter space; corresponding values of $q(\lambda)$ are computed; and these values are binned to produce an approximate probability distribution on the output.

The stochastic inverse sensitivity analysis problem that we study is the inversion of the *Law of Total Probability* (2.2).

> *We assume that an observed probability density $\rho_\mathcal{D}(q(\lambda))$ is given on the output value $q(\lambda)$, and we seek to compute the corresponding parameter density $\sigma_\Lambda(\lambda)$ that yields $\rho_\mathcal{D}(q(\lambda))$ via* (2.2).

It is important to note that what we seek for the solution of the inverse problem is the actual probability density that can be used to compute the probability of events in the parameter space Λ. In other words, we seek to compute the inverse of the analogue of the Perron–Frobenius map between the densities on the input and output spaces. The purpose of this paper is to describe a method for solving the inverse problem by providing a way to approximate the probability of an arbitrary event in the input space. This can be used subsequently to generate an approximation of the inverse density and/or to compute any desired statistical moments of the inverse density.

We emphasize the fundamental role of the underlying parameter volume measure μ_Λ in defining the solution of the inverse problem. In particular, the a priori specification of μ_Λ imposes the structure of the measure on Λ, e.g., whether the measure on Λ is a product measure or not. In general, there are many combinations of σ_Λ and μ_Λ that can yield a given observed density on the output.

We provide a simple illustration of the inverse problem using the map

$$q(\lambda) = \lambda_1 + \lambda_2,$$

where λ_1, λ_2 are random variables. For the inverse problem, we specify that $q(\lambda)$ has a $N(0, 2/25)$ distribution and seek to determine the parameter distribution $\sigma_\Lambda(\lambda)$ that yields the specified output density. This output distribution can be generated by choosing λ_1, λ_2 to be independent identically distributed $N(0, 1/25)$ random variables; see Figure 2.1. As well, we could choose any bivariate normal density

$$\begin{pmatrix}\lambda_1\\ \lambda_2\end{pmatrix} \sim N\left(\begin{pmatrix}-\alpha\\ \alpha\end{pmatrix}, \tau^2\begin{pmatrix}1 & \varrho\\ \varrho & 1\end{pmatrix}\right) \text{ with } 2\tau^2(1+\varrho) = \frac{2}{25}, \quad \varrho \in [-1, 1].$$

If we find a distribution on Λ that generates $q(\lambda)$ according to a $N(0, 2/25)$ distribution, then we accept this as a solution to the inverse problem. The choice of the underlying parameter volume measure μ_Λ is critical to this task. In Figures 2.1–2.3, we show five different probability densities $\sigma_\Lambda(\lambda)$ that yield the identical $N(0, 2/25)$ density on $q(\lambda)$. Each of the five different densities correspond to five different underlying volume distributions μ_Λ as shown.

The specification of μ_Λ has to do with how measurements in Λ are carried out and the relationships between the parameters. As noted, the volume measure should be specified as part of defining the model. In many situations involving deterministic models, the product Lebesgue measure appropriately scaled to account for units is the natural choice. But, this is not always the case. Continuing the motivating problem, as a first approximation, we might consider the thickness and alloy composition to be physically independent parameters and impose a product measure on the space formed by the two variables using independent normalized Lebesgue measures. A

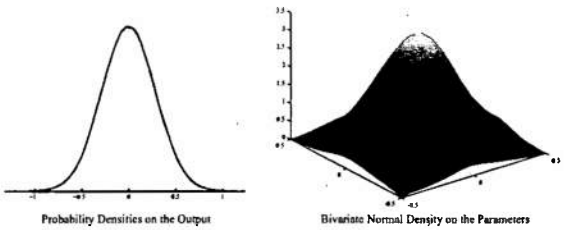


FIG. 2.1. *Left: The $N(0, 2/25)$ distribution imposed on the output $\lambda_1 + \lambda_2$. Right: The joint distribution of two independent $N(0, 1/25)$ parameters λ_1 and λ_2. Summing these variables is one way to compute the imposed normal on the output quantity. Figures 2.2–2.3 show alternatives.*

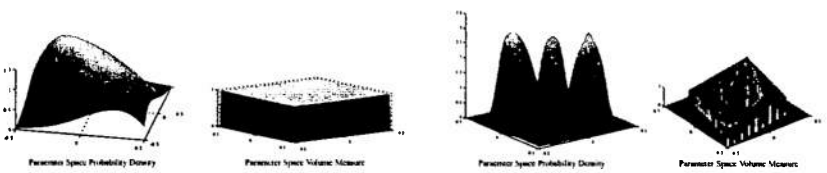


FIG. 2.2. *The joint distributions of parameters (λ_1, λ_2) sampled with respect to the density $\rho_\Lambda(\lambda)$ and the corresponding volume measure presented in pairs of plots. Left two plots: The volume measure is uniform Lebesgue on Λ. Right two plots: The volume measure is uniform Lebesgue a set with three distinct parts.*

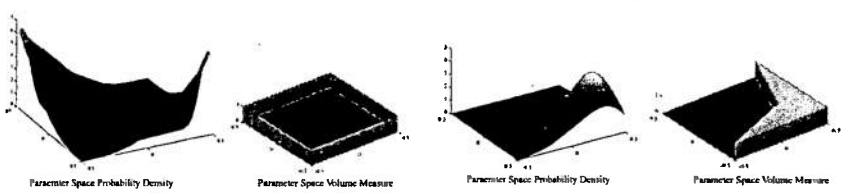


FIG. 2.3. *The joint distributions of parameters (λ_1, λ_2) sampled with respect to the density $\rho_\Lambda(\lambda)$ and the corresponding volume measure presented in pairs of plots. Left two plots: The volume measure is uniform Lebesgue on the boundary. Right two plots: The volume measure is uniform Lebesgue on a nonconvex interior set.*

more realistic description will take into account the fact that the thickness of the plates indirectly depends on the alloy composition during the milling process. We can model the milling process to determine the thickness as an indirect function of the physically independent variables of pressure in the milling process and the alloy composition. The measure on the space consisting of the thickness and alloy composition is then determined by propagating the product measure imposed on the independent alloy composition and pressure variables through the milling model. The resulting measure on the space consisting of the alloy composition and thickness will not be a product measure.

The plots of inverse densities given in Figures 2.2–2.3 also illustrate the important point that injecting probability into the inverse problem by itself does not reduce the ill-posedness, even after specifying the parameter volume measure. The consequence of ill-posedness on the stochastic inverse problem is illustrated by the complex measure structure of the inverse probability densities in the plots. For example, these densities are not product measures. In general, it is not possible to determine densities for the

individual parameters without further information. We can determine only a measure on the entire parameter space.

Comparison to a Bayesian inverse problem. There is another natural inverse problem associated with the *Law of Total Probability* (2.1) that is important in the case of a *general* likelihood function $L(q|\lambda)$, not necessarily arising from a deterministic map. Namely, we may use Bayes' theorem to invert the likelihood function to obtain the "posterior density" $p(\lambda|q)$ given the "prior density" σ_Λ on the input space Λ and a "data density" $\rho_\mathcal{D}$ on the output space $\mathcal{D}$. We emphasize that the solution of this Bayesian inverse problem is a conditional distribution. This is very natural when the map from the input to output space has been modeled statistically by specifying $L(q|\lambda)$ given information about the statistical properties of the input parameters and output quantity, e.g., when the map is derived empirically, rather than from physical principles.

This Bayesian inverse problem is at the heart of Bayesian inference [26, 1, 19, 18]. In this approach, the inferential target is a single, unknown parameter (or parameter vector) λ. We are given data in the form of observations $q_1, \ldots, q_n$, for which a typical assumption is conditional independence,

$$p(q_1, \ldots, q_n \,|\, \lambda) \sim \prod_{i=1}^{n} p(q_i \,|\, \lambda), \tag{2.3}$$

where $\{p(q_i \,|\, \lambda)\}$ are conditional probability densities with respect to some appropriate measure, and are specified up to the value of λ. The right-hand side of (2.3) is the likelihood of the observations given the parameter. We are also given a prior distribution on λ that gives a probabilistic description of the uncertainty about the values of λ before any data are observed. This prior distribution is exactly $\sigma_\Lambda(\lambda)$ in the notation used above. Bayesian inference then proceeds by using Bayes' theorem to compute the posteriori conditional distribution of λ given the observations $q_1, \ldots, q_n$:

$$p(\lambda \,|\, q_1, \ldots, q_n) \propto p(q_1, \ldots, q_n \,|\, \lambda)\, \sigma_\Lambda(\lambda) = \prod_{i=1}^{n} p(q_i \,|\, \lambda)\, \sigma_\Lambda(\lambda). \tag{2.4}$$

We could adopt a Bayesian approach to solve the inverse problem we study by modeling $\sigma_\Lambda(\lambda)$ parametrically as $\sigma_\Lambda(\lambda \,|\, \theta)$ in terms of new (lower-dimensional) parameters θ. This is known as a *mixture* or *hierarchical* model. In Bayesian terminology, $\sigma_\Lambda(\lambda \,|\, \theta)$ is the prior while a new distribution σ_θ describing θ is the hyperprior. Assuming that the hyperprior is specified, we then compute the posterior distribution on θ given "data" from $\rho_\mathcal{D}(q(\lambda))$. Any desired inferences about the distribution of λ given θ can then be obtained from the posterior. The difficulty with this approach is specifying a reasonable conditional model, which is difficult to verify empirically.

The inverse problem solved in this paper shares some characteristics with the Bayesian inverse problem, but has fundamental differences as well. In the Bayesian problem, the inferential target is the *parameter* λ, and σ_Λ is *given* as prior information. The likelihood $L(q \,|\, \lambda)$ typically involves a nontrivial stochastic structure and is *not deterministic.*

By contrast, in the inverse problem we solve the inferential target is the *distribution* σ_Λ, which is *not given* as the prior. Further, our likelihood $L(q \,|\, \lambda)$ is given by a *deterministic* map, which completely determines the set-valued inverse.

The choice of inverse problem to solve depends completely on the available information. In the case of a deterministic physics-based model, the unknowns and

quantities subject to uncertainty are the data and parameter values that are input into the model and the observations that are supposed to match model output while the likelihood function determined by the map is completely trivial in a statistical/probabilistic sense. Based on the law of total probability, the inverse problem we solve is the direct inverse of the probabilistic forward sensitivity problem for a deterministic model.

3. Solving the inverse problem. As noted above, while probability densities describe the random nature of a random variable, the densities themselves are not random. While a common approach to compute a discrete approximation of a probability density employs random sampling, this is not necessary. In this paper, we describe a method for computing approximate probability densities that does not require random sampling. Our approach breaks the solution down into two stages:

1. Construct an approximate representation of the set-valued inverse solution of the deterministic model.
2. Use measure-theoretic computational methods to approximate the probability density (measure) structure on the parameter space that corresponds to the set-valued inverse and the observed output density.

These are independently interesting tasks.

We present a brief overview before providing the details. Under the assumption of a smooth map, if we are given a fixed output value $\bar{q} \in \mathcal{D}$, then the implicit function theorem guarantees the existence of a $(d-1)$-dimensional manifold in Λ that is mapped to $\bar{q}$. Motivation comes from the two-dimensional case, $\lambda = (\lambda_1, \lambda_2)$, where the manifolds are contours of the surface $q(\lambda_1, \lambda_2)$ (left-hand illustration in Figure 3.1). Every point in Λ lies on a unique contour, so we may consider Λ as a set described by its contours. The set of (generalized) contours is an equivalence class in the input space, i.e., a quotient space representation of the input space. In Λ, there exists 1-dimensional curves transverse to the contours that intersect each contour once and only once (right-hand illustration in Figure 3.1). We can take one of these curves as the index for the set of contours. There is a bijection between the points on an index curve and the points in the range of the output $q(\Lambda)$. Therefore, any measure posed on the range of the output imposes a measure on the index curve. Thus, the intersections of the contours with the index curve is a random variable with

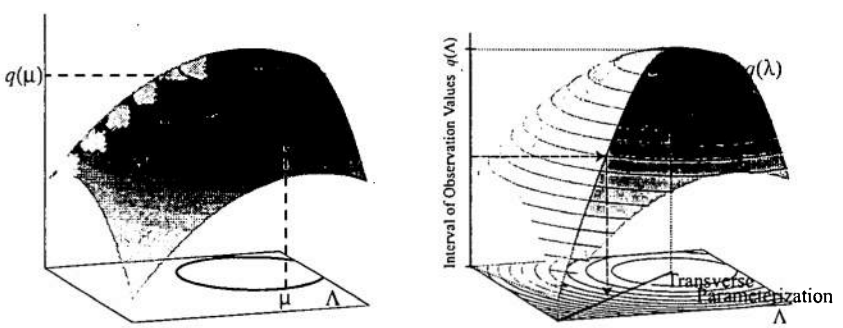


FIG. 3.1. *Left: Each observation value corresponds to a unique contour curve. Right: On the horizontal plane, we show a transverse parameterization. Each point on the transverse parameterization corresponds to a unique contour curve, so the transverse parameterization acts as an index for the space of contour curves. There is a unique map from the points in the interval containing the observed output values to the points on the transverse parameterization.*

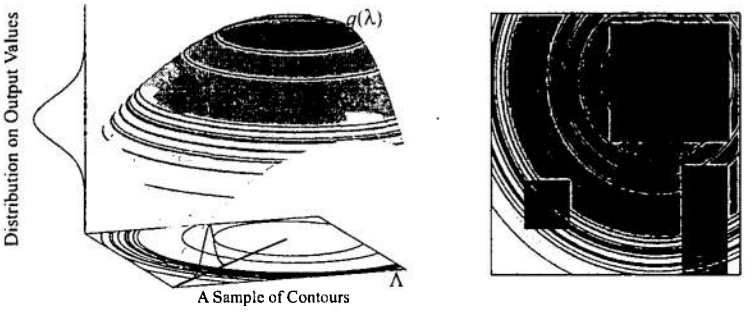


FIG. 3.2. *Left: We show a probability distribution imposed on the output values. A sample of output values drawn from this distribution corresponds to a unique sample of contour curves. Right: Plotted is a sample of contour lines in parameter space corresponding to a specified distribution on the output observation values along with three events. We specify the Lebesgue measure as the parameter volume measure. Event B has relatively low probability because while it has relatively large area, the probability of the contours is relatively low (visible because the density is sparse). Event A has intermediate probability because while the area of event A is relatively small, A contains contours with relatively high probability (which is visible because of the dense sample of contours). The probability of event C is largest because it contains the same high probability contours as A but has larger area.*

a distribution uniquely defined by the distribution of the output $\rho_{\mathcal{D}}(q(\lambda))$ (left-hand illustration in Figure 3.2). In other words, there exists a unique solution to the inverse sensitivity analysis problem in the set of the contours.

However, determining the set of contours analytically is infeasible in practice. In [23], the forward sensitivity analysis problem defined by (2.2), where a given density $\sigma_{\Lambda}(\lambda)$ is propagated through the output surface $q(\lambda)$, is solved using a piecewise-linear tangent plane approximation to the output surface. This requires computations involving only inner products, which is cheap compared to the full model evaluation cost of $q(\lambda)$ for each new value of λ. The derivatives of $q(\lambda)$ are computed implicitly using adjoint methods. Motivated by this approach, we use a piecewise-linear tangent plane approximation to the output surface $q(\lambda)$ to construct approximate contours and an approximate index set.

The next step is to determine the probability density on the parameter set that corresponds to the distribution on the transverse parameterization of the space of approximate contours. In order to assign a probability to a measurable set in Λ, we first recognize that such a set is defined by the contours it contains and the amount of each contour it contains (right-hand illustration in Figure 3.2). The parameter volume measure μ_{Λ} specified on Λ quantifies the amount of each contour contained in any given set. Combining the results of the generalized contours with such a measure, the monotone convergence theorem, and additivity properties of measures, we develop an algorithm to estimate the probability of any measurable set in Λ. This algorithm employs a piecewise constant approximation of measures that is commonly used in measure theory. This yields a direct computational method to approximate $\sigma_{\Lambda}(\lambda)$.

In the next two sections, we provide details of the two ingredients of the approximate solution method.

Remark 3.1. Many solution methods for both statistical and deterministic inverse problems deal with ill-posedness by introducing some form of regularization, either directly or reposing the inverse problem as an optimization problem. Such methods avoid the need to deal with set-valued inverse solutions.

Remark 3.2. There are cases of interest, e.g., a parameter domain that contains a bifurcation point, for which the described method cannot be used in a straightforward fashion. We note that while an approach based on random sampling may be applied nominally to such a problem, the interpretation of the results is still problematic.

Remark 3.3. While the solution method for the inverse problem proposed here relies on derivatives of a quantity of interest, it is not dependent on how those derivatives are computed. Instead of an adjoint-based approach, the derivatives might be computed using (deterministic) forward sensitivity analysis that computes the derivatives directly along with the solution of the model. Yet another approach, presented, e.g., in [27], employs a stochastic spectral method to obtain a polynomial representation of $q(\lambda)$, which is then used to compute gradients.

3.1. Determining the inverse of the deterministic model using generalized contours. We consider a finite dimensional map q from the space of parameters to the output defined implicitly by solving a finite dimensional nonlinear system of equations,

$$f(x;\lambda) = b, \tag{3.1}$$

where $x \in \mathbb{R}^n$, parameter $\lambda \in \mathbf{\Lambda} \subset \mathbb{R}^d$ (assuming that $\mathbf{\Lambda}$ is compact) is a random vector, and $f : \mathbb{R}^{n+d} \to \mathbb{R}^n$ is assumed smooth in both variables. The goal is to compute a quantity of interest $q(\lambda) = q(x(\lambda)) = \langle x, \psi \rangle$, described as a linear functional of the solution $x(\lambda)$. If x depends smoothly on λ, then the dependence of q on λ is also smooth.

This problem applies in particular to differential equations that depend on a finite set of parameters. For differential equations, we require the same assumptions as the standard existence and uniqueness theorems to guarantee the smoothness of $q(\lambda)$. This is discussed in more detail in the second part of this paper [4].

For any $\bar{q} \in q(\mathbf{\Lambda})$, we define $\check{q}(\lambda) := q(\lambda) - \bar{q}$. By assumption, $\check{q}(\lambda) : \mathbb{R}^d \to \mathbb{R}$ is continuously differentiable and there exists $\bar{\lambda} \in \mathbf{\Lambda}$ such that $q(\bar{\lambda}) = \bar{q}$, which implies that $\check{q}(\bar{\lambda}) = 0$. We are mainly interested in the case where the quantity of interest varies as the parameters vary, so we assume that $\partial_{\lambda_d} \check{q}(\bar{\lambda}) \neq 0$, i.e. there is at least one nontrivial partial derivative. We may relax the restriction of $\partial_{\lambda_d} \check{q}(\bar{\lambda}) \neq 0$ for a finite number of points in $\mathbf{\Lambda}$, where $q(\lambda)$ possibly attains a local extreme value and ignore this set of points when considering the generalized contours.

By the implicit function theorem, there exists an open set $U_{\bar{\lambda}} \subset \mathbf{\Lambda}^{d-1}$, where $\mathbf{\Lambda}^{d-1} := \{\lambda^{d-1} := (\lambda_1, \ldots, \lambda_{d-1}) | \lambda = (\lambda_1, \ldots, \lambda_d) \in \mathbf{\Lambda}\}$, containing $\bar{\lambda}^{d-1}$, an open set $V_{\bar{\lambda}} \subset \mathbf{\Lambda}_d$, where $\mathbf{\Lambda}_d := \{\lambda_d | \lambda \in \mathbf{\Lambda}\}$, and a differentiable function $g_{\bar{\lambda}} : U_{\bar{\lambda}} \to V_{\bar{\lambda}}$ such that

$$\{(\lambda^{d-1}, g_{\bar{\lambda}}(\lambda^{d-1}))\} = \{\lambda | q(\lambda) = \bar{q}\} \cap (U_{\bar{\lambda}} \times V_{\bar{\lambda}}). \tag{3.2}$$

Since the implicit function theorem is a local result, there may be additional points in $\mathbf{\Lambda}$ that map to $\bar{q}$, but are not contained in the set defined by (3.2). Thus, given $\bar{q} \in q(\mathbf{\Lambda})$, we choose a collection of sets $\{U_{\bar{\lambda}} \times V_{\bar{\lambda}}\} = \bigcup_{\alpha \in A} \{U_{\bar{\lambda}_\alpha} \times V_{\bar{\lambda}_\alpha}\}$, where $\bigcup_{\alpha \in A} \{\bar{\lambda}_\alpha\}$ is the set of all $\lambda \in \mathbf{\Lambda}$ such that $q(\lambda) = \bar{q}$. Then using the same notation as in (3.2), the function $g_{\bar{\lambda}}(\lambda^{d-1})$ might be piecewise defined. The set in (3.2) is a $(d-1)$-dimensional manifold that is a natural inverse of $q(\lambda)$ given $\bar{q}$. We call this set the *generalized contour*.

THEOREM 3.1. *If we choose distinct $\bar{q}_1, \bar{q}_2 \in q(\Lambda)$, then the generalized contours for $\bar{q}_1$ and $\bar{q}_2$ are unique and do not intersect.*

Proof. The nonintersection property follows immediately from the fact that $q(\lambda)$ is a function. Uniqueness follows immediately from the choice $\{U_{\bar{\lambda}} \times V_{\bar{\lambda}}\} = \bigcup_{\alpha \in A}\{U_{\bar{\lambda}_\alpha} \times V_{\bar{\lambda}_\alpha}\}$, where $\bigcup_{\alpha \in A}\{\bar{\lambda}_\alpha\}$ is the set of all $\lambda \in \Lambda$ such that $q(\lambda) = \bar{q}$ for a given value of $\bar{q} \in q(\Lambda)$. □

In two dimensions, the generalized contours are simply contours of the surface $q(\lambda_1, \lambda_2)$. We denote a generalized contour for a specific quantity of interest $\bar{q}$ as $q^{-1}(\bar{q})$. Since $q(\lambda)$ is smooth and Λ is compact, $q(\Lambda)$ defines a compact interval of real numbers, $I_q := [q_m, q_M] = q(\Lambda)$, where q_m and q_M are the absolute minimum and absolute maximum of $q(\lambda)$, respectively. We redefine $q(\Lambda)$ to be the open interval (q_m, q_M), which we also denote by I_q.

We next prove that there exists (possibly discontinuous) 1-dimensional curves that are transverse to the generalized contours that can be used to *index* the family of generalized contours. We call any curve that has the property that it intersects each generalized contour once and only once a *transverse parameterization* (TP).

We give a constructive proof that is a useful algorithm. The algorithm produces discontinuous curves in Λ in general.

THEOREM 3.2. *Suppose f is smooth in* (3.1) *and $q(\lambda)$ is a linear functional of the solution to* (3.1). *There exists a transverse parameterization for the set of generalized contours.*

Proof. We construct the transverse curve from a finite number of connected curves. We fix $\epsilon > 0$ and $\epsilon > \delta > 0$, and set $I_{q,\epsilon} = [q_m + \epsilon, q_M - \epsilon]$. If Λ is compact, then the existence of transverse curves is guaranteed by the smoothness of $q(\lambda)$. To construct a curve, we begin at a point $\gamma_M \in \Lambda$ such that $q(\gamma_M) = q_M - \delta$, and follow the direction of the negative gradient until the curve either intersects the boundary or a minimum or saddle is reached, and denote that point γ_m. From smoothness, exactly one contour for each value of $q(\lambda)$ between $(q(\gamma_m), q(\gamma_M))$ is intersected by this curve. If $(q(\gamma_m), q(\gamma_M))$ does not completely cover $I_{q,\epsilon}$, then we select a point $\tau_m \in \Lambda$ such that $q(\tau_m) = q_m + \delta$, and follow the direction of the gradient until the curve either intersects the boundary or a maximum or saddle is reached, and denote this point τ_M. We now check if $(q(\gamma_m), q(\gamma_M)) \cup (q(\tau_m), q(\tau_M))$ covers $I_{q,\epsilon}$. If so, then we eliminate any part of the second curve that gives an overlap with contours intersected by the first. Otherwise, we continue to create this curve as above trying to cover the output interval defined by $(q(\tau_M), q(\gamma_m))$. This process produces a countable number of connected curves whose union forms a (possibly discontinuous) transverse curve through the generalized contours that corresponds to a countable open cover of $I_{q,\epsilon}$, which is compact. Hence, there is a finite subcover of $I_{q,\epsilon}$, which implies that the transverse parameterization can be constructed from a finite number of curves. □

In practice, we construct the transverse curve to the generalized contours of I_q by initially following the first two steps above with $\epsilon = 0$, i.e., locate $\gamma_M \in \Lambda$ such that $q(\gamma_M) = q_M$ and $\tau_m \in \Lambda$ such that $q(\tau_m) = q_m$ and construct the pieces of the transverse curve by following the negative and positive directions of the gradient, respectively. If we now take ϵ to be half the minimum of $q(\gamma_M) - q(\gamma_m)$ and $q(\tau_M) - q(\tau_m)$, then following the steps above, we construct a curve transverse to all the contours of I_q in a finite number of steps.

3.1.1. Approximating the set of generalized contours. Suppose that q is a linear function of λ, i.e., $q(\lambda) = \gamma^T\lambda$ for some $\gamma \in \mathbb{R}^d$ (recall $\Lambda \subset \mathbb{R}^d$). Then for fixed $\bar{q} \in q(\Lambda)$ we have (with the same conventions as above) $U_{\bar{\lambda}}$, $V_{\bar{\lambda}}$, and $g_{\bar{\lambda}} : U_{\bar{\lambda}} \to V_{\bar{\lambda}}$ such

that $\{(\lambda^{d-1}, g_{\bar{\lambda}}(\lambda^{d-1}))\}$ is the generalized contour. In this case, we write the function $g_{\bar{\lambda}}(\lambda^{d-1}) = (\bar{q} - (\gamma^{d-1})^{\top}(\lambda^{d-1}))/\gamma_d$ explicitly. The generalized contour above is a $(d-1)$-dimensional hyperplane, and we refer to this as a *generalized linear contour.*

We approximate generalized contours locally by generalized linear contours, and approximate a generalized contour by a generalized piecewise-linear contour. We use generalized piecewise-linear contours computed from a piecewise-linear tangent plane approximation to $q(\lambda)$. If q is an affine map of λ, i.e., $q(\lambda) = \gamma^T\lambda + q_0$ for some $q_0 \in \mathbb{R}$, then we use the function above with $\bar{q}$ replaced by $\bar{q} - q_0$.

We obtain derivative information required to compute the tangent plane approximations implicitly by introducing the adjoint operator. This approach is very useful when the forward map is complicated to evaluate, e.g., involving the solution of a differential equation. But, the derivative information can be obtained by any convenient method.

Local linearization of the linear functional. The goal is to approximate the map $q(\lambda)$ with a piecewise-linear map $\hat{q}(\lambda)$ since it is possible to calculate the generalized contours for this approximate map.

THEOREM 3.3. *The generalized linear contours converge pointwise to the true contours locally in* $\mathbf{\Lambda}$.

Proof. Suppose we choose a reference parameter value $\lambda = \mu$ at which to solve

$$f(x;\lambda) = b$$

exactly. Call this reference solution y. Then according to Taylor's theorem,

$$f(x;\lambda) = f(y;\mu) + D_x f(y;\mu)(x-y) + D_\lambda f(y;\mu)(\lambda-\mu) + \mathcal{R},$$

where $\mathcal{R} \sim O(\|x-y\|^2 + \|\lambda-\mu\|^2)$, for $|\alpha| = 2$. Here $D_x f$ and $D_\lambda f$ denote the derivatives of f with respect to x and λ, respectively.

In order to compute the tangent plane approximation efficiently, we use the *generalized Green's vector* ϕ that solves the adjoint to the linearized problem

$$A^{\top}\phi = \psi, \tag{3.3}$$

where $A = D_x f(y;\mu)$. Recall that $q(\lambda) = \langle x, \psi\rangle$, so by substitution of the above and standard linear algebra we arrive at

$$q(\lambda) = q(\mu) - \langle D_\lambda f(y;\mu)(\lambda-\mu), \phi\rangle - \langle \mathcal{R}, \phi\rangle .$$

Neglecting the higher order term leads to an approximation of q by an affine map $\hat{q}$. If we denote the generalized contour of q given $\bar{q}$ by $\{(\lambda^{d-1}, g_{\bar{\lambda}}(\lambda^{d-1}))\}$ and the generalized linear contour of $\hat{q}$ given $\bar{q}$ by $\{(\lambda^{d-1}, \hat{g}_{\bar{\lambda}}(\lambda^{d-1}))\}$, then at any $\lambda^{d-1} \in U_{\bar{\lambda}}$,

$$\left[g_{\bar{\lambda}}(\lambda^{d-1}) - \hat{g}_{\bar{\lambda}}(\lambda^{d-1})\right]\left[\phi^{\top}\partial_{\lambda_d} f(y,\mu)\right] = -\langle \mathcal{R}, \phi\rangle . \tag{3.4}$$

By assumption, $\partial_{\lambda_d} q(\lambda) = \phi^{\top}\partial_{\lambda_d} f(y,\mu) \neq 0$, so we rewrite (3.4) as

$$\left[g_{\bar{\lambda}}(\lambda^{d-1}) - \hat{g}_{\bar{\lambda}}(\lambda^{d-1})\right] = C\langle \mathcal{R}, \phi\rangle ,$$

where $C^{-1} = -\phi^{\top}\partial_{\lambda_d} f(y,\mu)$, is a nonzero constant determined entirely by the reference point (y,μ). Thus, if we define

$$\|U_{\bar{\lambda}}\| = \sup_{\lambda \in U_{\bar{\lambda}}} \|\lambda - \bar{\mu}\|_2 ,$$

where $\|\ \|_2$ denotes the standard Euclidean norm, then as $\|U_{\bar{\lambda}}\| \to 0$, $\|\mathbf{R}\|_2 \to 0$, which implies that $\left|g_{\bar{\lambda}}(\lambda^{d-1}) - \hat{g}_{\bar{\lambda}}(\lambda^{d-1})\right| \to 0$. □

Global linearization of the linear functional. We extend the local linearization technique to obtain a global piecewise-linear approximation of the linear functional over all of $\mathbf{\Lambda}$. We first define a partition of *cells* $\{B_i\}_{i=1}^M$ of $\mathbf{\Lambda}$. The geometry is immaterial, as long as we can integrate constant functions over the cells. We apply the local linearization technique described above for each cell, and defining

$$\mathbf{1}_{B_i}(\lambda) := \begin{cases} 1 & \text{if } \lambda \in B_i, \\ 0 & \text{if } \lambda \notin B_i, \end{cases}$$

we obtain a global piecewise-linear approximation $\hat{q}(\lambda)$ to $q(\lambda)$ defined by

$$\hat{q}(\lambda) := \sum_{i=1}^{M} \left(q(\mu_i) + \langle \nabla q(\mu_i), (\lambda - \mu_i) \rangle \right) \mathbf{1}_{B_i}(\lambda), \tag{3.5}$$

where μ_i is the reference parameter value chosen in cell B_i.

THEOREM 3.4. *As* $\|B_i\| \to 0$ *(or as* $M \to \infty$ *when the number of sample points are distributed uniformly), the generalized linear contour converges pointwise to the generalized contour.*

Proof. For the finite system of nonlinear equations, we have

$$\nabla q(\mu_i) = \phi_i^\top D_\lambda f(y_i; \mu_i),$$

where ϕ_i solves the linearized adjoint problem using the reference point (y_i, μ_i). If we let $-\langle \mathcal{R}_i, \phi_i \rangle$ denote the higher-order terms neglected in the linearization of $q(\lambda)$ in cell B_i, then we can write the error of the piecewise-linear approximation, $e(\lambda) = \hat{q}(\lambda) - q(\lambda)$, as

$$e(\lambda) = -\sum_{i=1}^{M} \langle \mathcal{R}_i, \phi_i \rangle \mathbf{1}_{B_i}(\lambda).$$

The generalized linear contour of $\hat{q}$ given $\bar{q}$ is a collection of hyperplanes in $\mathbf{\Lambda}$. Using the same notation as above,

$$\left| g_{\bar{\lambda}}(\lambda^{d-1}) - \hat{g}_{\bar{\lambda}}(\lambda^{d-1}) \right| \le C \sum_{i=1}^{M} \left| \langle \mathcal{R}_i, \phi_i \rangle \right|, \qquad C^{-1} = \min_i \left\{ \left| \phi_i^\top \partial_{\lambda_d} f(y_i, \mu_i) \right| \right\}.$$

This yields the convergence result. □

The transverse parameterization (TP) for the generalized linear contours is constructed using $\hat{q}$ in the same way as described in the proof of Theorem 3.2. Since $\hat{q}$ is a piecewise-linear surface, the resulting TP is a piecewise-linear curve in $\mathbf{\Lambda}$.

Examples. We illustrate the convergence of generalized linear contours to true contours in the two examples below.

In the first example, we suppose that $q(\lambda_1, \lambda_2) = \lambda_1 \lambda_2 \exp\left[-(\lambda_1^2 + 1.25\lambda_2^2 - 1)\right]$ over $[0, 2] \times [0, 2]$. We approximate q over a uniform partition $\{B_i\}$ of $[0, 2] \times [0, 2]$ into squares, and we linearize around the midpoint of each B_i to form $\hat{q}$ in (3.5). We plot various contour curves and two TP's on each plot. The results are summarized in Figure 3.3.

For a second example, we suppose $q(\lambda_1, \lambda_2) = \exp\left[\cos(\lambda_1) + \sin(\lambda_2)\right]$ on $[-2\pi - 0.1, 2\pi + 0.1]^2$. We proceed as above to obtain the numerical results summarized in Figure 3.4.

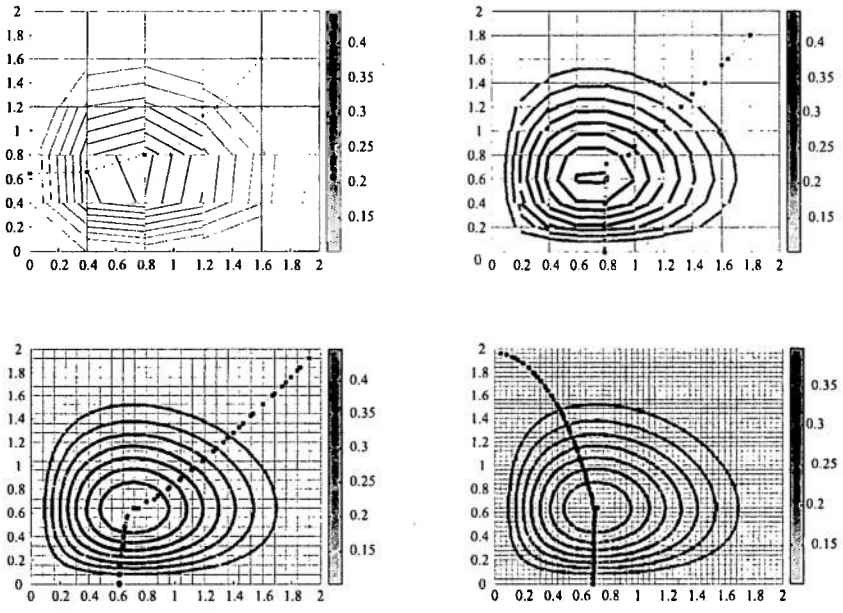


FIG. 3.3. *Contours of $\hat{q}$ using 5×5 cells (top left), 10×10 cells (top right), 25×25 cells (bottom left), and 50×50 cells (bottom right). The TP is created using the algorithm outlined in the proof of its existence and is denoted by the circle-dotted and plus-dotted lines. The circle-dotted line is constructed from the maximum of $q(\lambda)$ and follows the negative direction of the gradient of $q(\lambda)$, and the plus-dotted line is constructed from the minimum of $q(\lambda)$ and follows the direction of the gradient.*

3.2. Computing the parameter probability density. We now explain how to use the unique solution to the inverse problem in the space of generalized contours to compute an approximation of the probability density σ_Λ on Λ. We first observe if $I = [q_1, q_2] \subset \mathcal{D}$ is an event with probability $P(I) = P(q(\lambda) \in I)$, then this corresponds to a measurable set in Λ that is defined as the set of all contours obtained by $q^{-1}(I)$. From the basic assumptions of smoothness and the nonintersecting property of the contours, the set of all contours is a set in Λ that is contained between the two contours defined by $q^{-1}(q_1)$ and $q^{-1}(q_2)$ (or possibly one of these contours and the boundary of Λ). We assign this set the probability $P(I)$. It follows immediately that we can define the inverse into the set of generalized contours for a given distribution of $q(\lambda)$ uniquely.

THEOREM 3.5. *Suppose f is smooth in* (3.1) *and $q(\lambda)$ is a linear functional of the solution to* (3.1). *If $q(\lambda)$ is a random variable with distribution $F_q(q(\lambda))$, then for a fixed TP in Λ, the distribution of the intersections of the generalized contours on the TP, which is a random variable, is unique.*

The probability of a measurable set in Λ is determined by the contours the set contains and the amount of each contour the set contains and the probabilities of those contours. The parameter volume measure μ_Λ determines the contours a given set contains and the amount of each contour the set contains.

3.2.1. Computational measure theory. The method we develop for computing an approximate probability distribution is based on constructions used in measure theory.

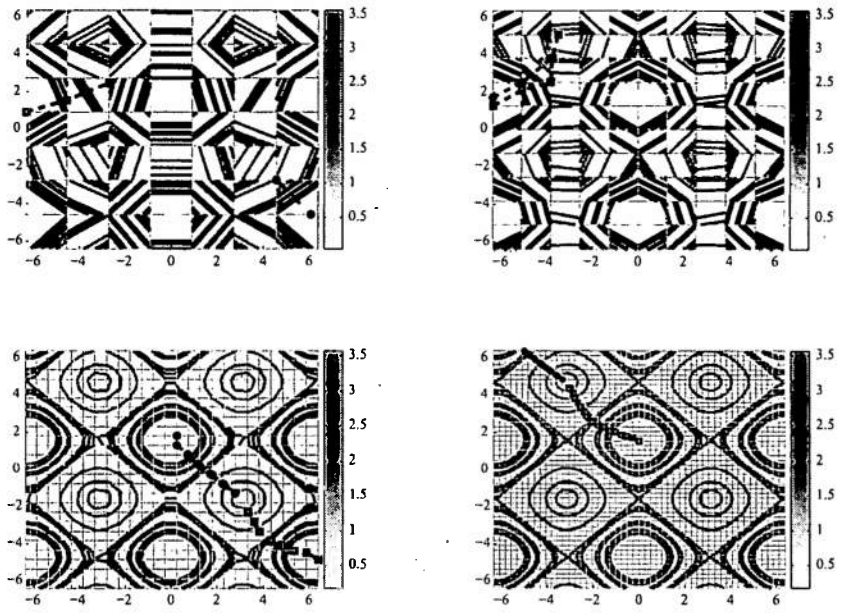


FIG. 3.4. *Contours of $\hat{q}$ using 7×7 cells (top left), 10×10 cells (top right), 25×25 cells (bottom left), and 50×50 cells (bottom right). The TP is created using the algorithm outlined in the proof of its existence and is denoted by the square-dotted and circle-dotted lines. The square-dotted line is constructed from the maximum of $q(\lambda)$ and follows the negative direction of the gradient of $q(\lambda)$, and the circle-dotted line is constructed from the minimum of $q(\lambda)$ and follows the direction of the gradient.*

THEOREM 3.6. *Given a measurable set $A \subset \Lambda$, we can approximate $P(A)$ using a simple function approximation to $\sigma_\Lambda(\lambda)$, which requires only calculations of volumes in Λ.*

The constructive proof below parallels Algorithm 1 for approximating the probability of a measurable set $A \subset \Lambda$.

Proof. For λ restricted between any two contours induced by a subinterval of a partition of $\mathcal{D}$ as in Algorithm 1, $q(\lambda)$ is approximately a uniformly distributed random variable. Suppose that $\{q_j\}_{j=0}^N$ is a partition of $\mathcal{D}$ such that $q_0 < q_1 < \cdots < q_N$, and if $E_j = [q_{j-1}, q_j]$, then $\mathcal{D} = \cup_j E_j$. Let $A_j = \{\lambda \,|\, q(\lambda) \in E_j\}$. We assume that $\Lambda = \cup_j A_j$. The probability of A_j is given by

$$P(A_j) = \int_{A_j} \sigma_\Lambda(\lambda)\, d\mu_\Lambda(\lambda).$$

We can compute this probability because of the 1-1 correspondence between the contours and output values, i.e., $P(A_j) = P(E_j) = \int_{E_j} \rho_\mathcal{D}(q)\, d\mu_\mathcal{D}(q)$. Therefore, we have a simple function approximation to $\sigma_\Lambda(\lambda)$ given by

$$\sigma_\Lambda(\lambda) \approx \sigma_{\Lambda,N}(\lambda) = \sum_{j=1}^{N} \frac{P(A_j)}{\mu_\Lambda(A_j)} \mathbf{1}_{A_j}(\lambda).$$

Algorithm 1. APPROXIMATE PARAMETER PROBABILITY DISTRIBUTION METHOD

Fix simple function approximation, $\rho_{\mathcal{D}}^{(M)}(q)$, to $\rho_{\mathcal{D}}(q)$ that induces a partition $\cup_{i=1}^{N(M)}[q_{i-1}, q_i)$ of $\mathcal{D}$ where for each $i = 1, \ldots, N(M)$, $\rho_{\mathcal{D}}^{(M)}(q)$ is constant on each subinterval $[q_{i-1}, q_i)$

$\cup_{i=1}^{N(M)}[q_{i-1}, q_i)$ induces a partition of $\mathbf{\Lambda}$ by generalized contours and $\{A_j\}_{j=1}^{N(M)}$ denotes this partition

Let P_j denote probability of A_j given by $\int_{[q_{j-1},q_j)} \rho_{\mathcal{D}}^{(M)}(q)\, d\mu_{\mathcal{D}}(q)$

Partition $\mathbf{\Lambda}$ with small cells $\{b_i\}_{i=1}^{M'}$

for $i = 1, \ldots, M'$ **do**

  for $j = 1, \ldots, N(M)$ **do**

    Calculate ratio of volume of $b_i \cap A_j$ to volume of A_j, store in matrix V_{ij}

  end for

  Set $P(b_i)$ equal to $\sum_{j=1}^{N(M)} V_{ij} P_j$

end for

Given event $A \subset \mathbf{\Lambda}$, estimate $P(A)$ using

- inner sums, i.e., sum of $P(b_i)$ for all $i \in I \subset \{1, \ldots, M'\}$ such that $b_i \subset A$,
- outer sums, i.e., sum of $P(b_i)$ for all $i \in I \subset \{1, \ldots, M'\}$ such that $b_i \cap A \neq \emptyset$,
- average of inner and outer sums, or
- $\int_A \sigma_{\mathbf{\Lambda},M'}(\lambda)\, d\mu_{\mathbf{\Lambda}}(\lambda)$, where $\sigma_{\mathbf{\Lambda},M'}(\lambda) = \sum_{i=1}^{M'} P(b_i)\mathbf{1}_{b_i}(\lambda)$.

Given event $A \subset \mathbf{\Lambda}$, we use the law of total probability to write

$$P(A) = \sum_{j=1}^{N} P(A \,|\, A_j) P(A_j).$$

Using the above simple function approximation to the parameter density, we have

$$P(A \,|\, A_j) = \frac{P(A \cap A_j)}{P(A_j)} = \frac{\int_{A \cap A_j} d\mu_{\mathbf{\Lambda}}(\lambda)}{\int_{A_j} d\mu_{\mathbf{\Lambda}}(\lambda)} = \frac{\mu_{\mathbf{\Lambda}}(A \cap A_j)}{\mu_{\mathbf{\Lambda}}(A_j)}.$$

Hence, the probability $P(\lambda \in A \,|\, q(\lambda) \in E_j) = P(A \,|\, A_j)$ can be calculated from the volume measure on model space since it depends only on measurable sets in $\mathbf{\Lambda}$ if we use the approximation $q(\lambda) \sim \mathcal{U}(E_j)$ for $\lambda \in A_j$. The value is the ratio of volume of $A \cap A_j$ to the volume of A_j. Since the density on data space is a nonnegative integrable function, there exists a sequence of simple functions $\{\rho_{\mathcal{D}}^{(M)}(q)\}_{M=1}^{\infty}$ with

$$\rho_{\mathcal{D}}^{(M)}(q) = \sum_{k=1}^{2^{2M}+1} \frac{k-1}{2^M} \mathbf{1}_{I_{M,k}}(\rho_{\mathcal{D}}(q)),$$

and $I_{M,k} = [(k-1)/2^M, k/2^M]$. We first observe that the partition $\{I_{M,k}\}$ induces a partition $\{E_{M,k}\}$ of $\mathcal{D}$. Also, we observe that $\rho_{\mathcal{D}}^{(M)}(q) \to \rho_{\mathcal{D}}(q)$ in L^1 as $M \to \infty$ by the monotone convergence theorem, and for any measurable set $E \subset \mathcal{D}$,

$$\int_E \rho_{\mathcal{D}}^{(M)}(q)\, d\mu_{\mathcal{D}}(q) = \sum_{k=1}^{2^{2M}+1} \frac{k-1}{2^M} \mu_{\mathcal{D}}(E_{M,k} \cap E) \to P_{\mathcal{D}}(E) \text{ as } M \to \infty.$$

Thus, we can approximate the value of $P(A \mid A_j)$ by the ratio of volume of $A \cap A_j$ to the volume of A_j obtained from the volume measure on model space if the induced partitions $\{A_j\}$ come from a sufficiently fine partition $\{E_j\}$ of data space so that the distribution of $q(\lambda)$ for $\lambda \in A_j$ is approximated by $\mathcal{U}(E_j)$.

Since $P(A) = \sup\{P(K) : K \subset A,\ K \text{ compact}\}$ and $P(A) = \inf\{P(U) : A \subset U,\ U \text{ open}\}$, we can estimate $P(A)$ using the inner and outer sums described by Algorithm 1. □

Remark 3.4. If the set A has not (yet) been specified, we may still carry out the first part of Algorithm 1 to obtain a *discretized* approximation of the measure P on model space.

Remark 3.5. The set of cells $\{b_i\}_{i=1}^{M'}$ in Algorithm 1 is introduced purely for computational purposes and is not necessary to the approximation of $P(A)$. We choose $\{b_i\}_{i=1}^{M'}$ in order to approximate $P(A)$, for *any* event $A \subset \mathbf{\Lambda}$, without carrying out the calculations in the nested loops of Algorithm 1 for each new event. If we are interested only in one event, $A \subset \mathbf{\Lambda}$, then we might skip the step of partitioning $\mathbf{\Lambda}$ by $\{b_i\}_{i=1}^{M'}$ and replace the step in the nested loop by the following: Calculate ratio of volume of $A \cap A_j$ to volume of A_j, store in vector V_j. We may then approximate $P(A)$ by $\sum_{j=1}^{N(M)} V_j P_j$.

Remark 3.6. Note that as we refine the partition $\{E_j\}$ on the data space, which in turn refines the partition $\{A_j\}$ on model space, we should consider refining the mesh that defines the partition $\{b_i\}$ on model space. The reason is that we assign a probability $P(b_i)$ to each cell b_i that in essence reapproximates the simple function approximation,

$$\sigma_{\mathbf{\Lambda}}(\lambda) \approx \sigma_{\mathbf{\Lambda},N}(\lambda) = \sum_{j=1}^{N} \frac{P(A_j)}{\mu_{\mathbf{\Lambda}}(A_j)} \mathbf{1}_{A_j}(\lambda),$$

by the new simple function

$$\sigma_{\mathbf{\Lambda}}(\lambda) \approx \sigma_{\mathbf{\Lambda},M'}(\lambda) = \sum_{i=1}^{M'} \frac{P(b_i)}{\mu_{\mathbf{\Lambda}}(b_i)} \mathbf{1}_{b_i}(\lambda).$$

If the partition $\{b_i\}$ remains fixed as the approximation of $\rho_{\mathcal{D}}(q)$ by simple functions is refined by the partition $\{E_j\}$, then the representation of $\sigma_{\mathbf{\Lambda}}(\lambda)$ as a simple function converges with respect to the fixed $\{b_i\}$. When choosing $\{b_i\}$, we should consider that a cell b_i might be large relative to the A_j that it intersects, i.e., b_i might intersect many A_j. When this is the case, estimating the probability over b_i by a constant $P(b_i)$ might not be an appropriate approximation. In general, it is not computationally demanding to estimate an appropriate size of the b_i.

Observations on simple function approximations. The use of simple function approximations of a probability density is sufficiently unusual in the context of stochastic analysis of differential equations as to justify comment. Simple function approximations form the basis for classic measure theory because they yield several benefits, including

- Simple function approximations are widely applicable under minimal assumptions on the density being approximated. As the examples below suggest, probability densities solving inverse problems appear to be highly complex in general.

- The convergence analysis for simple function approximations is also widely applicable. This contrasts with sampling techniques such as Markov chain Monte Carlo methods whose convergence properties are stochastic and can be highly sensitive to properties of the problem.

Though we have not exploited the fact in this paper, simple function approximations also offer significant benefits for stochastic sensitivity analysis of differential equations [12, 13, 9, 10, 7]. In particular, combining a simple function approximation with sensitivity derivatives of a quantity of interest with respect to parameters provides both a natural dimension reduction mechanism and the basis for adaptive sampling.

Of course, a significant issue with simple function approximations is the nominal dependence of accuracy on the dimension of the parameter space. This may be a consequence of the common approach of using hyper-rectangular cell discretizations of the underlying space combined with the unfortunate growth in diagonal dimension of hyper-rectangles as dimension increases, though we report on some inconclusive results of using radial basis functions in [12]. In our experience, the effects of dimension are nominal up to dimensions of 8–10, and we have effectively used the piecewise constant approximations to dimensions of order 15–18. We note that this is *effective* dimension. By exploiting dimension reduction, the nominal dimension of the parameter space may be higher.

4. Examples. We apply the new method to solve inverse problems associated with a variety of maps. We first consider three constrained geometric optimization problems. We then discuss examples involving a nonlinear ordinary differential equation and a nonlinear elliptic partial differential equation with two parameters. Finally, we discuss the determination of regions with high probability.

In the following examples, we have chosen the uniform Lebesgue measure for the parameter volume measure and often impose a normal distribution on the output quantity of interest. The first choice is made because it is commonly the (implicit) default, e.g., in Bayesian inference. The imposition of a normal distribution on the output is also a common choice. In our examples, it serves the purpose of illustrating the complex nature of the inverse probability measure that results even when a normal distribution has been imposed on the output. However, we emphasize that neither of these choices are important in terms of implementing the numerical solution method, which is readily applied for any distributions.

4.1. A 2-dimensional nonlinear function. We consider the map determined implicitly as the solution of the finite-dimensional nonlinear system of equations given by

$$\begin{aligned} \lambda_1 x_1^2 + x_2^2 &= 1, \\ x_1^2 - \lambda_2 x_2^2 &= 1, \end{aligned}$$

where λ_1 and λ_2 are the parameters. Geometrically, solutions $x = (x_1, x_2)^T$ to the system represent intersections of the hyperbola and ellipse. The quantity of interest is the second component of the solution in the first-quadrant, i.e., $q(\lambda) = q(x(\lambda)) = x_2 = \langle x, \psi \rangle$, where $\psi = (0, 1)^T$. According to (3.3), the adjoint problem is

$$\begin{pmatrix} 2\mu_1 y_1 & 2y_1 \\ 2y_2 & -2\mu_2 y_2 \end{pmatrix} \phi = \psi,$$

where $\mu = (\mu_1, \mu_2)^T$ and $\mathbf{y} = (y_1, y_2)^T$ are the reference parameter and reference solution for the forward problem.

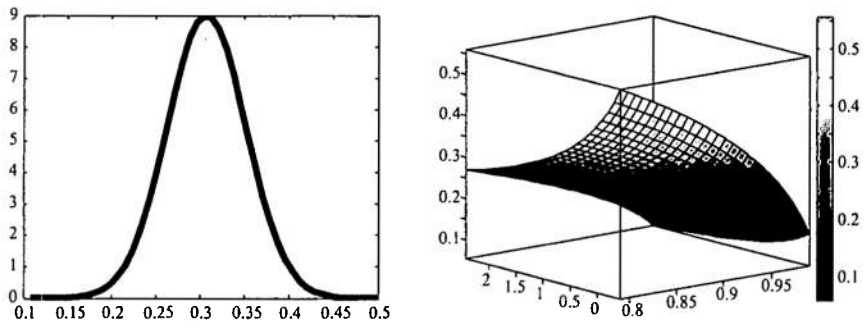


FIG. 4.1. *Left: Uncertainty of output is modeled as a random variable with a normal distribution. Right: A plot of the map* $q : \Lambda \to \mathbb{R}$.

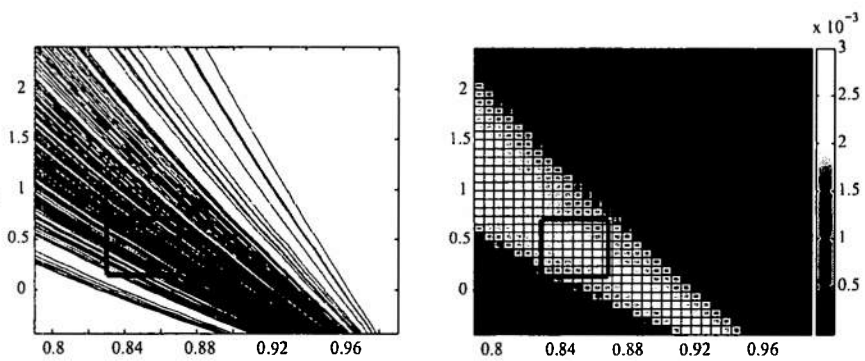


FIG. 4.2. *Illustration of an application of Algorithm* 1. *Left: We determine which contours are contained in an event* $A \subset \Lambda$ *and how much of each contour is inside the event. Right: We estimate the probabilities of small cells contained in the event and use an inner and outer estimate to obtain an approximation of the probability of the event* A.

In order to create an interesting example, we choose $\Lambda = [.79, .99] \times [1 - 4.5\sqrt{0.1}, 1 + 4.5\sqrt{0.1}]$ based on a sensitivity analysis of the forward problem in [23]. We use six-uniformly spaced mesh points in both the λ_1 and λ_2 directions of Λ to create cells $\{B_i\}_{i=1}^{25}$ that partition Λ. We use the centroid of each cell as the reference parameter $\mu_i = (\mu_{1,i}, \mu_{2,i})^T$ in that cell and solve the forward problem to obtain reference solutions $y_i = (y_{1,i}, y_{2,i})^T$ at these points, and then solve for the generalized Green's vector $\phi_i = (\phi_{1,i}, \phi_{2,i})^T$ at the reference point (μ_i, y_i). According to (3.5), we obtain a global piecewise-linear approximation $\hat{q}$ to q defined as

$$\hat{q}(\lambda) := \sum_{i=1}^{25} \left(y_{2,i} + (\lambda - \mu_i)^T \begin{pmatrix} y_{1,i}^2 & 0 \\ 0 & -y_{2,i}^2 \end{pmatrix} \phi_i \right) \mathbf{1}_{B_i}(\lambda).$$

We assume that the output data is a random variable with normal distribution on the data space defined by $\hat{q}(\Lambda)$ (Figure 4.1). We assume μ_Λ is the Lebesgue measure. We implement Algorithm 1 to calculate $P(b_i)$ for small cells for each fine partition of Λ and determine the probabilities of events $A \subset \Lambda$. We plot the results in Figure 4.2.

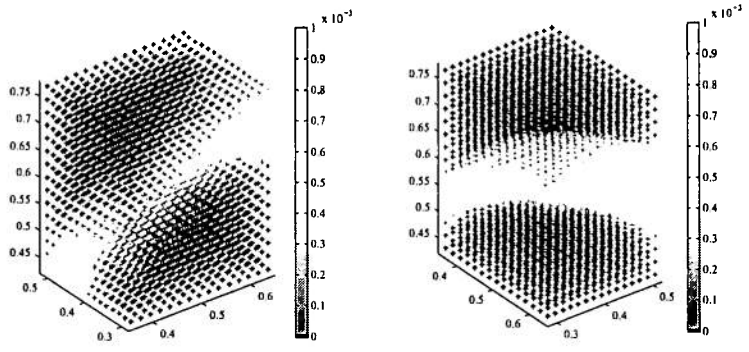


FIG. 4.3. *We use* $15 \times 15 \times 15$ *small cells in Algorithm* 1. *We plot the approximate distribution from several angles. Left: A* 3*-dimensional view. Right: The same* 3*-dimensional view rotated* 90 *degrees clockwise.*

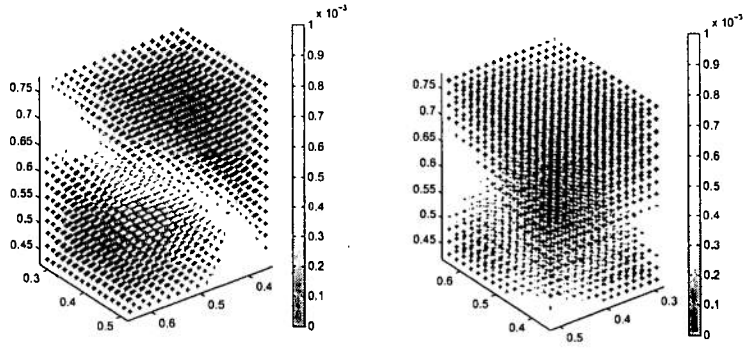


FIG. 4.4. *We use* $15 \times 15 \times 15$ *small cells in Algorithm* 1. *We plot the approximate distribution from several angles. Left: The original* 3*-dimensional view rotated* 180 *degrees clockwise. Right: The original* 3*-dimensional view rotated* 270 *degrees clockwise.*

4.1.1. A three-parameter geometric constrained optimization problem. The map to be inverted is determined by minimizing the distance to the point $(1, -1, 1)$ among points constrained to lie on the surface $g = 4$, where

$$g(x_1, x_2, x_3; \lambda_1, \lambda_2, \lambda_3) = \lambda_1 x_1^2 + \lambda_2 x_2^2 + \lambda_3 x_3^2.$$

Geometrically, the parameters determine the shape of the ellipsoid that defines the constraint. Using the method of Lagrange multipliers we set up a system of nonlinear equations with four state variables and three parameters. We take the quantity of interest as the first state variable, which geometrically is interpreted as the first spatial coordinate in the solution to the constrained minimization problem. We set $\mathbf{\Lambda} = [.35, .65] \times [.28, .52] \times [.42, .78]$ and construct a piecewise-linear approximation using 125 points in $\mathbf{\Lambda}$. We assume a normal distribution on $q(\lambda)$ and taking the underlying parameter volume measure $\mu_{\mathbf{\Lambda}}$ to be a normalized Lebesgue measure. We use 3375 small cells $\{B_i\}$ in Algorithm 1. We plot the probabilities at the midpoint of each cell with the color of the point determined by the probability of the small cell in Figures 4.3–4.4.

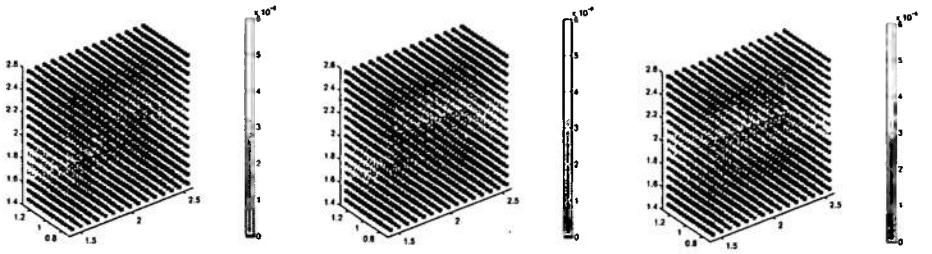


FIG. 4.5. *We use* $15 \times 15 \times 15 \times 18$ *small cells in Algorithm* 1. *We plot "snapshots" of the approximate probability distribution for three values of the fourth parameter. Left: The fourth parameter is set at its minimum value. Middle: The fourth parameter is set at its midpoint value. Right: The fourth parameter is set at its maximum value. Notice how the probabilities vary in space as we vary the fourth parameter.*

4.1.2. A four-parameter geometric constrained optimization problem. The map to be inverted is determined by minimizing the distance to the point $(5, 5, 5)$ among points constrained to lie on the intersection of the surfaces $g = 1$ and $h = 0$, where

$$g(x_1, x_2, x_3; \lambda_1, \lambda_2) = \lambda_1 x_1^2 + \lambda_2 x_2^2 - x_3^2,$$
$$h(x_1, x_2, x_3; \lambda_3, \lambda_4) = \lambda_1 x_1 + \lambda_2 x_2 - x_3.$$

Geometrically, $g = 1$ defines a hyperboloid of one sheet and $h = 0$ defines a plane through the origin, and the intersection of the two constraints is a closed curve. Using the method of Lagrange multipliers we set up a system of nonlinear equations with five state variables and four parameters. We take the quantity of interest as the first state variable, which geometrically is interpreted as the first spatial coordinate in the solution to the constrained minimization problem. We set $\mathbf{\Lambda} = [1.4, 2.6] \times [.7, 1.3] \times [1.4, 2.6] \times [.35, .65]$ and construct a piecewise-linear approximation using 750 points in $\mathbf{\Lambda}$. We assume a normal distribution on $q(\lambda)$ and take $\mu_{\mathbf{\Lambda}}$ to be a normalized Lebesgue measure. We use 60750 small cells $\{b_i\}$ in Algorithm 1. Displaying a 4-dimensional distribution is problematic. We plot "snapshots" of the approximated probability density for three fixed λ_4 values in Figure 4.5.

4.1.3. A two-parameter ordinary differential equation. We now study the nonlinear ordinary differential equation

$$\begin{cases} \dot{x} = \lambda_1 \sin(\lambda_2 x), & 0 < t \leq T, \\ x(0) = 1. \end{cases}$$

The linear functionals (quantities of interest, $q(\lambda)$) we study take the form

$$q(\lambda) = \langle x(t), \psi(t) \rangle = \int_0^T (x(s; \lambda), \psi(s))\, ds,$$

and we take the quantity of interest to be the average value of $x(t)$ over the time interval $[0, 2]$. Thus, we set $\psi(t) = \mathbf{1}_{[0,2]}(t)/2$, and the generalized Green's function $\phi(t)$ solves the adjoint problem,

$$\begin{cases} -\dot{\phi}(t) - A^{\top}(t)\phi(t) = \psi(t), & T > t \geq 0, \\ \phi(T) = \psi(T), \end{cases}$$

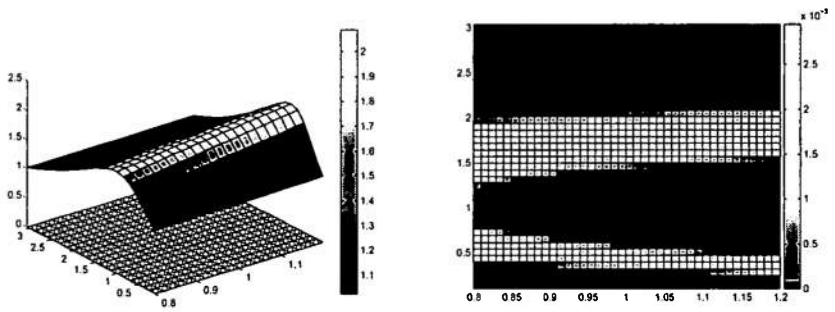


FIG. 4.6. *Left: The global piecewise-linear approximation to $q(\lambda)$ obtained using Algorithm 1. The cells in Λ illustrate the coarse discretization of this space for the forward problem of obtaining a piecewise-linear approximation and the circles in each cell indicate the reference parameter used to linearize $q(\lambda)$ in that cell. We assume a normal distribution for $q(\lambda)$ and use a grid of 40×40 small cells.*

where $A(t) := f'(y(t;\mu))$ is the Jacobian of $f = \lambda_1 \sin(\lambda_2 x)$ evaluated at $y(t;\mu)$, μ is a reference parameter, and $y(t;\mu)$ is the solution to (4.1.3) for this reference parameter. Compare this to (3.3). Using substitution, integration by parts, and Taylor's theorem, we arrive at a linear approximation to $q(\lambda)$ for parameters near μ, and analogous to the finite dimensional case, we obtain a global piecewise-linear approximation to $q(\lambda)$ over $\Lambda = [.8, 1.2] \times [.1, \pi - .1]$ shown in Figure 4.6.

Remark 4.1. There can be substantial error in the reference solutions and gradients used when applying the method to differential equations whose solutions must be approximated numerically, and we study the effect of these errors in the second paper [4].

4.1.4. A two-parameter elliptic partial differential equation. We now study a nonlinear elliptic partial differential equation

$$\begin{cases} -\Delta u = \lambda_1 (u - \lambda_2)^2, & (x,y) \in \Omega = [0,1] \times [0,1], \\ u = 0, & (x,y) \in \partial\Omega. \end{cases}$$

The quantities of interest, $q(\lambda)$, take the form

$$q(\lambda) = \langle u, \psi \rangle = \int_\Omega u(x,y)\psi(x,y)\,dxdy,$$

and we take the quantity of interest to be the average value of u over Ω. Thus, we set $\psi(x,y) = 1$, and the generalized Green's function $\phi(t)$ solves the adjoint problem,

$$\begin{cases} -\Delta\phi - A^\top\phi = \psi, & (x,y) \in \Omega, \\ \phi = 0, & (x,y) \in \partial\Omega, \end{cases}$$

where $A := f'(w(x,y;\mu);\mu)$ is the Jacobian of $f = \lambda_1 \exp(\lambda_2 u)$ evaluated at $w(x,y;\mu)$, μ is a reference parameter, and $w(x,y;\mu)$ is the solution to (4.1.4) for this reference parameter. Using substitution, the weak form of (4.1.4), and Taylor's theorem, we arrive at a linear approximation to $q(\lambda)$ for parameters near μ, and just as with the previous examples, we obtain a global piecewise-linear approximation to $q(\lambda)$ over $\Lambda = [.95, 1.05] \times [-.1, .1]$ using Algorithm 1. We show the results in Figure 4.7.

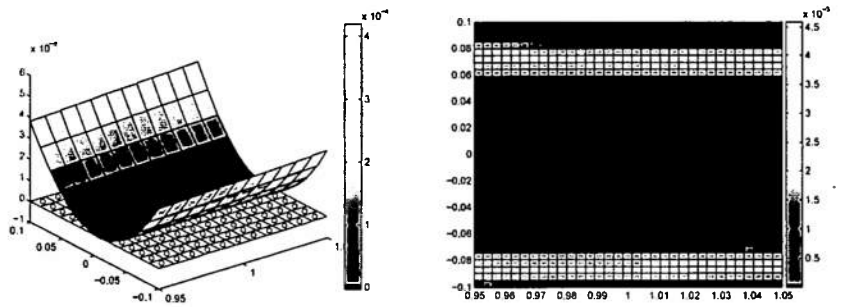


FIG. 4.7. *Left: Global piecewise-linear approximation to $q(\lambda)$ obtained using Algorithm* 1. *We used a* 11×13 *grid of coarse cells to discretize* Λ *and used the midpoint of each cell as the reference parameter in that cell. We assume a normal distribution of* $q(\lambda)$ *and we use a* 33×39 *grid of small cells.*

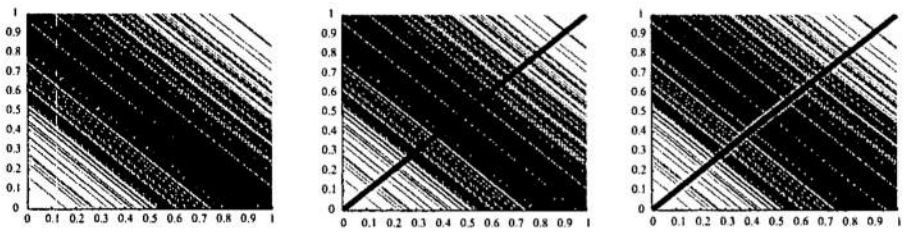


FIG. 4.8. *Left: Generalized contours from* 500 *samples of* $q(\lambda) = \lambda_1 + \lambda_2$ *generated from a* $N(0, 2/25)$ *distribution. Middle: The TP intersects each contour once and goes from the minimum of* $q(\lambda)$ *in the lower left corner to the maximum of* $q(\lambda)$ *in the upper-right corner of the plot. Right: Intersections of contours on the TP are marked with a star and can be used to index the inverses and determine a unique distribution of the contours on the TP using any consistent indexing scheme*

4.2. Determining regions of high probability. The new method can be applied to find regions of high probability. Consider $q(\lambda) = \lambda_1 + \lambda_2$, where $\Lambda = [0,1] \times [0,1]$. Figure 4.8 shows the generalized contours for 500 samples of $q(\lambda)$ taken from a $N(0, 2/25)$ distribution along with the TP and the intersections of contours on the TP. Where the contours intersect the TP most densely corresponds to a region of high probability in the space of contours.

We can locate regions of high probability by sorting through the probability of the fine cells $\{b_i\}$. We can rank order these cells and determine any cells of high probability. We can also determine regions of neighboring cells that all have relatively high probability. We illustrate using the four-parameter geometric constrained optimization problem in section 4.1.2. In Table 1, we list the ten small cells with highest probability. If we let the events $\{b_i\}$ become small, under a smoothness assumption, the probabilities of these events are related to the maximum-likelihood estimate.

5. Conclusion. We consider the probabilistic inverse sensitivity analysis problem: Given a specified uncertainty in the output of a map, determine variations in the parameters that produce the observed uncertainty. We formulate this inverse problem using the law of total probability. We describe and analyze a method for computing the approximate probability density that solves the inverse problem and does not require random sampling. Our approach breaks the solution down into two stages:

TABLE 1

We indicate the location of the ten cells with the highest probabilities for the example in section 4.1.2. The first column gives the probability and the second column gives the dimensions and location of the cells. There are clearly two distinct regions for events with relatively high probability. In general, one can use this information to determine where the largest regions of highest probability are located in a high-dimensional parameter space.

$P(b_i)$ order 10^{-4}	b_i location
0.600381927	$[2.44, 2.52] \times [1.22, 1.26] \times [2.04, 2.12] \times [0.4, 0.4167]$
0.600446977	$[2.36, 2.44] \times [1.06, 1.1] \times [1.96, 2.04] \times [0.4333, 0.45]$
0.600462420	$[2.44, 2.52] \times [1.18, 1.22] \times [2.04, 2.12] \times [0.4333, 0.45]$
0.600465732	$[2.36, 2.44] \times [0.98, 1.02] \times [2.04, 2.12] \times [0.4167, 0.4333]$
0.600470136	$[2.36, 2.44] \times [1.06, 1.1] \times [1.96, 2.04] \times [0.4167, 0.4333]$
0.600474821	$[2.36, 2.44] \times [1.26, 1.3] \times [1.96, 2.04] \times [0.4167, 0.4333]$
0.600501752	$[2.36, 2.44] \times [0.98, 1.02] \times [2.04, 2.12] \times [0.4333, 0.45]$
0.600463048	$[1.4, 1.48] \times [1.18, 1.22] \times [1.64, 1.72] \times [0.3833, 0.4]$
0.600464252	$[1.4, 1.48] \times [1.18, 1.22] \times [1.64, 1.72] \times [0.35, 0.3667]$
0.600468545	$[1.4, 1.48] \times [1.18, 1.22] \times [1.64, 1.72] \times [0.3667, 0.3833]$

1. Construct an approximate representation of the set-valued inverse solution of the ill-posed deterministic inverse problem.
2. Approximate the density on the parameter space that corresponds to the set-valued inverse and the observed output density using a simple function representation.

We illustrate the method and several features using a variety of examples.

In [4] we present numerical analysis of discretization error, e.g., in evaluating the model by numerical solution and in finite sampling. In [5], we discuss the problem of dealing with multiple quantities of interest, which has application to data assimilation and "cascaded" uncertainty in operator decomposition solution of multiphysics problems.

REFERENCES

[1] J.M. BERNARDO, *Reference posterior distributions for Bayesian inference*, J. Roy. Statist. Soc. Ser. B, 41 (1979), pp. 113–147.

[2] P. BILLINGSLEY, *Probability and Measure*, 3rd ed., John Wiley & Sons, New York, 1995.

[3] T. BUTLER, *Computational Measure Theoretic Approach to Inverse Sensitivity Analysis: Methods and Analysis*, Ph.D. thesis, Department of Mathematics, Colorado State University, Fort Collins, CO, 2009.

[4] T. BUTLER AND D. ESTEP, *A measure-theoretic computational method for inverse sensitivity problems* II: *A posterior error analysis*, SIAM J. Numer. Anal., submitted.

[5] T. BUTLER AND D. ESTEP, *A measure-theoretic computational method for inverse sensitivity problems* III: *Multiple output quantities of interest*, in preparation, 2010.

[6] D. CACUCI, *Sensitivity and Uncertainty Analysis: Theory*, Vol. I, Chapman & Hall/CRC, Boca Raton, FL, 1997.

[7] D. ESTEP, M.J. HOLST, AND A. MÅLQVIST, *Nonparametric density estimation for randomly perturbed elliptic problems* III: *Convergence, complexity, and generalizations*, J. Appl. Math. Comput., to appear.

[8] D. ESTEP, M.G. LARSON, AND R.D. WILLIAMS, *Estimating the error of numerical solutions of systems of reaction-diffusion equations*, Mem. Amer. Math. Soc., 146 (2000), pp. viii+109.

[9] D. ESTEP, A. MÅLQVIST, AND S. TAVENER, *Nonparametric density estimation for randomly perturbed elliptic problems*. I: *Computational methods, a posteriori analysis, and adaptive error control*, SIAM J. Sci. Comput., 31 (2009), pp. 2935–2959.

[10] D. ESTEP, A. MÅLQVIST, AND S. TAVENER, *Nonparametric density estimation for randomly perturbed elliptic problems*. II: *Applications and adaptive modeling*, Internat. J. Numer. Methods Engrg., 80 (2009), pp. 846–867.

[11] D. ESTEP, B. MCKEOWN, D. NECKELS, AND J. SANDELIN, *GAASP: Globally Accurate Adaptive Sensitivity Package*, 2006, write to estep@math.colostate.edu for information.

[12] D. ESTEP AND D. NECKELS, *Fast and reliable methods for determining the evolution of uncertain parameters in differential equations*, J. Comput. Phys., 213 (2006), pp. 530–556.

[13] D. ESTEP AND D. NECKELS, *Fast methods for determining the evolution of uncertain parameters in reaction-diffusion equations*, Comput. Methods Appl. Mech. Engrg., 196 (2007), pp. 3967–3979.

[14] J.P. HUELSENBECK, B. LARGET, R.E. MILLER, AND F. RONQUIST, *Potential applications and pitfalls of Bayesian inference of phylogeny*, Syst. Biol., 51 (2002), pp. 673–688.

[15] G. FOLLAND, *Real Analysis*, John Wiley & Sons, Modern Techniques and their Applications, 2nd ed., New York, 1999.

[16] J.E. GENTLE, *Random Number Generation and Monte Carlo Methods*, 2nd ed., Springer, New York, 2003.

[17] W.R. GILKS, S. RICHARDSON, AND D.J. SPIEGELHALTER, *Markov Chain Monte Carlo in Practice*, CRC Press, Boca Raton, FL, 1995.

[18] J. KAIPID AND E. SOMERSALD, *Statistical and Computational Inverse Problems*, Springer-Verlag, New York, 2005.

[19] D.C KNILL AND W. RICHARDS, *Perception as Bayesian Inference*, Cambridge University Press, Cambridge, UK, 1996.

[20] C. LANCZOS, *Linear Differential Operators*, Dover Publications, Mineola, NY, 1997.

[21] G.I. MARCHUK, *Adjoint Equations and Analysis of Complex Systems*, Kluwer Academic Publishers, Dordrecht, 1995.

[22] G.I. MARCHUK, V.I. AGDSHKOV, AND V.P. SHUTYAEV, *Adjoint Equations and Perturbation Algorithms in Nonlinear Problems*, CRC Press, Boca Raton, FL, 1996.

[23] D. NECKELS, *Variational Methods for Uncertainty Quantification*, Ph.D. thesis, Department of Mathematics, Colorado State University, Fort Collins, CO, 2005.

[24] C.P. ROBERT AND G. CASELLA, *Monte Carlo Statistical Methods*, Springer-Verlag, New York, 2004.

[25] J. SANDELIN, *Global Estimate and Control of Model, Numerical, and Parameter Error*, Ph.D. thesis, Department of Mathematics, Colorado State University, Fort Collins, CO, 2006.

[26] A. TARANTOLA, *Inverse Problem Theory and Methods for Model Parameter Estimation*, SIAM, Philadelphia, 2005.

[27] N. ZABARAS AND B. GANAPATHYSUBRAMANIAN, *A scalable framework for the solution of stochastic inverse problems using a sparse grid collocation approach*, J. Comput. Phys., 227 (2008), pp. 4697–4735.

BLOCKWISE ADAPTIVITY FOR TIME DEPENDENT PROBLEMS BASED ON COARSE SCALE ADJOINT SOLUTIONS

V. CAREY *, D. ESTEP †, A. JOHANSSON ‡, M. LARSON §, AND S. TAVENER ¶

Abstract. We describe and test an adaptive algorithm for evolution problems that employs a sequence of "blocks" consisting of fixed, though non-uniform, space meshes. This approach offers the advantages of adaptive mesh refinement but with reduced overhead costs associated with load balancing, re-meshing, matrix reassembly, and the solution of adjoint problems used to estimate discretization error and the effects of mesh changes. A major issue with a block-adaptive approach is determining block discretizations from coarse scale solution information that achieve the desired accuracy. We describe several strategies to achieve this goal using adjoint-based *a posteriori* error estimates and we demonstrate the behavior of the proposed algorithms as well as several technical issues in a set of examples.

Key words. *a posteriori* error analysis, adaptive error control, adaptive mesh refinement, adjoint problem, discontinuous Galerkin method, duality, generalized Green's function, goal oriented error estimates, residual, variational analysis

AMS subject classifications. 65N15, 65N30, 65N50

1. Introduction. We describe and test an adaptive algorithm for evolution problems that we call "blockwise adaptivity". This approach employs a sequence of "blocks" consisting of fixed, though non-uniform, space meshes, and is motivated by considerations of efficiency and accuracy. We balance the goal of achieving desired accuracy using discretizations with relatively few degrees of freedom against the computational costs associated with load balancing, re-meshing, matrix reassembly and in particular the cost of error estimation. A block adaptive strategy reduces the number of mesh changes that must be treated, which reduces the amount of computational time spent on re-meshing, assembly, and load balancing, and makes the problem of quantifying the effects of mesh changes on accuracy computationally feasible. A block adaptive strategy also provides a natural coarse scale discretization on which to solve the adjoint problem used to compute global *a posteriori* error estimates. This reduces the twin computational difficulties of storing a fine scale forward solution in order to form the adjoint problem and solving the adjoint problem on that fine scale discretization. However, a major issue with a block-adaptive approach is determining block discretizations from coarse scale solution information that achieve

*Department of Mathematics, Colorado State University, Fort Collins, CO 80523 (carey@math.colostate.edu). V. Carey's work is supported in part by the Department of Energy (DE-FG02-04ER25620)

†Department of Mathematics and Department of Statistics, Colorado State University, Fort Collins, CO 80523 (estep@math.colostate.edu). D. Estep's work is supported in part by the Defense Threat Reduction Agency (HDTRA1-09-1-0036), Department of Energy (DE-FG02-04ER25620, DE-FG02-05ER25699, DE-FC02-07ER54909, DE-SC0001724), Lawrence Livermore National Laboratory (B573139, B584647), the National Aeronautics and Space Administration (NNG04GH63G), the National Science Foundation (DMS-0107832, DMS-0715135, DGE-0221595003, MSPA-CSE-0434354, ECCS-0700559), Idaho National Laboratory (00069249), and the Sandia Corporation (PO299784)

‡Department of Mathematics, Umea University, S-90187 Umea, Sweden (august.johansson@math.umu.se). A. Johansson's work is supported by Umea University

§Department of Mathematics, Umea University, S-90187 Umea, Sweden (mats.larson@math.umu.se). M. Larson's work is supported in part by the Swedish Science Foundation (5730 33 150)

¶Department of Mathematics, Colorado State University, Fort Collins, CO 80523 (tavener@math.colostate.edu). S. Tavener's work is supported in part by the Department of Energy (DE-FG02-04ER25620)

the desired accuracy and efficiency. We describe several strategies to achieve this goal using adjoint-based *a posteriori* error estimates.

To focus the discussion, we consider a reaction-diffusion equation for the solution u on an interval $[0,T]$,

$$\begin{cases} \dot{u} - \nabla \cdot (\epsilon(x,t)\nabla u) = f(u,x,t), & (x,t) \in \Omega \times (0,T], \\ u(x,t) = 0, & (x,t) \in \partial\Omega \times (0,T], \\ u(x,0) = u_0(x), & x \in \Omega, \end{cases} \tag{1.1}$$

where Ω is a convex polygonal domain in $\mathbb{R}^d$ with boundary $\partial\Omega$, $\dot{u}$ denotes the partial derivative of u with respect to time, and there is a constant $\epsilon > 0$ such that

$$\epsilon(x,t) \geq \epsilon, \quad x \in \Omega,\ t > 0.$$

We also assume that ϵ and f have smooth second derivatives. The algorithms in this paper generalize to problems with different boundary conditions, convection, nonlinear diffusion coefficients, as well as systems, see [17, 15].

In terms of adaptive mesh refinement, the interesting situation is a solution of (1.1) that exhibits "regionalized" behavior in space and time. Considerations of efficiency suggests that time steps and space meshes should be locally refined to match the regional behavior, see the plot on the left in Fig. 1.1. Classic adaptive mesh refinement can be described as a constrained optimization problem, e.g., determine a discretization using the fewest degrees of freedom that yields a solution satisfying a given error criterion. In general, it is impossible to determine a closed-form solution of this optimization problem. An adaptive algorithm is an iterative procedure for determining a nearly optimal solution.

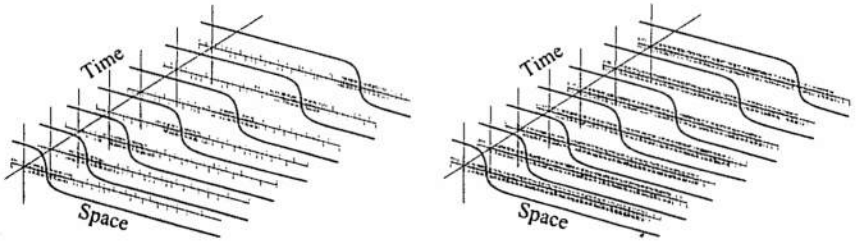


FIG. 1.1. *The evolution of a traveling front solution. Left: A computation using space meshes chosen by a standard adaptive strategy to control the spatial residual error at each time step. This entails re-meshing, re-assembly, load balancing, and projecting the solution on a new mesh at each step. Right: The uniform mesh that is required to achieve the same control over the residual. The computation is assembled and load balanced only once.*

We present a generic adaptive algorithm in Algorithm 1.1. An adaptive computation is generally started with an initial coarse mesh. The adaptive algorithm is then applied "real-time" as the integration proceeds so as to generate a new space mesh for each new time step, where the new space mesh is based on (or adapted from) the mesh for the current time step. In practice, the remeshing may be applied on intervals of a small number of steps.

While adaptive mesh refinement is appealing on an intuitional level, there are serious issues facing its use for evolution problems including the following.

Algorithm 1.1 Generic Adaptive Algorithm for an Evolution Problem

```
Choose an initial coarse mesh and time step
while the final time has not been reached do
    Compute a numerical solution using the current time step and space mesh
    Estimate the error of the computed solution
    while the error estimate is too large do
        Estimate local error contributions and adapt in space
        Estimate local error contributions and adapt in time
        Compute a numerical solution using the new time step and space mesh
        Estimate the error of the computed solution
    end while
    Increment time by the accepted time step
end while
```

1. ***Accuracy*** Each spatial mesh change requires a projection of the numerical solution onto the new mesh, and this can affect accuracy. In fact, this can destroy convergence altogether, see [8].
2. ***Overhead Costs*** Changing the spatial discretization requires generating a new mesh and reassembly of matrices. Significant mesh changes require a redistribution of unknowns among the processors to achieve load balancing. All of these tasks are computationally intensive.
3. ***Coarsening*** Un-refinement or coarsening of a mesh involves loss of information about a numerical solution that cannot be recovered. Currently, there is no theory for coarsening that guarantees that there is no loss of accuracy.
4. ***Global Error Estimation*** Efficient adaptive mesh refinement requires accurate error estimates of the true, global error, but cancelation of errors over both space and time makes choosing adapted meshes problematic.

Using a fixed spatial mesh eliminates the first three issues. But, the scale required of the mesh is determined by the finest scale required in any region where discretization impacts global accuracy, see Fig. 1.1. This necessarily increases computational time and solver costs and memory limits may make it impossible to use the necessary uniform mesh.

In this paper, we propose a "blockwise" adaptive algorithm that employs nonuniform meshes that remain fixed for discrete period of times, or "blocks", see Fig. 1.2. With the proper implementation, this strategy addresses the following key issues.

1. ***Accuracy*** The projections onto new meshes occur at a relatively small set of discrete times. We use *a posteriori* error estimates to predict the effect of the projections and choose overlaps in the meshes to reduce the error induced by the mesh changes.
2. ***Overhead Costs*** Re-meshing, assembly, and load balancing are required only at the discrete times demarcating blocks.
3. ***Coarsening*** There is ***no*** coarsening of a given mesh in the indicated strategy. Mesh changes are handled purely as projections between different meshes.

The idea of re-meshing only after a fixed number of steps is by no means new. However, this strategy depends critically upon choosing suitable block discretizations, and thus, ultimately, on accurately predicting the behavior of the solution. The choice of block discretizations is a difficult issue that requires balancing the inefficiency of using a fixed spatial mesh inside each block against the gain in accuracy achieved

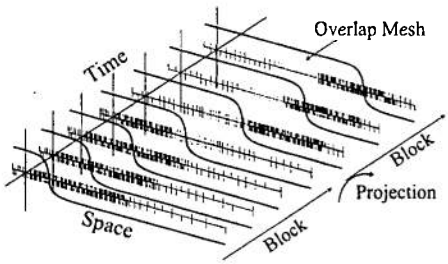


FIG. 1.2. *The evolution of a solution with a traveling front computed using blockwise adaptivity with two blocks. On each block, the space mesh is chosen to maintain the same level of control over the local residual as is achieved in the computation shown in Fig. 1.1. In addition, there is a sufficient degree of overlap between the two meshes (the lightly-shaded mesh region) to insure there is no loss of accuracy in projecting the solution between the two meshes. Re-meshing, assembly, and load balancing is only required twice, once for each block.*

by limiting projections between different meshes and the decrease in computational cost due to limiting the number of times at which re-meshing, re-assembly, and load balancing is required. This is partly a computer science problem of distributing available resources, e.g., memory and compute cycles, efficiently, and partly a numerical analysis problem, e.g., determining meshes for each block and projections between blocks.

In this paper, we focus on the problem of determining blocks, e.g., the length of times for each block, the meshes for each block that maintain accuracy in the desired information, and suitable overlap meshes for transitions between blocks from the coarse-scale adjoint solutions. The solutions of these problems require accurate estimates of the error in a specific quantity of interest. We use a computable *a posteriori* error estimate that yields robustly accurate estimates of the error in a specified quantity of interest in terms of a sum of space-time element contributions, see [9, 10, 17, 15, 3, 20]. The *a posteriori* error estimates are based on duality, adjoint problems, and variational analysis. Accurate error estimates are obtained by numerically solving the linear adjoint problem related to the desired quantity of interest.

Solving adjoint problems offers computational challenges such as the need to store the forward solution in order to form the adjoint problem and the cost of the adjoint solve. Our approach is to perform the adjoint solves using relatively coarse scale discretizations and using a coarse scale representation of the forward solution to form the adjoint problem, which reduces the memory overhead and the cost of the adjoint solve. This approach is motivated by the following observations.

1. Adjoint problems are linear and often present fewer numerical difficulties than the associated forward problems.
2. Solutions of adjoint problems tend to vary slowly on the scale of the discretization, whereas residuals of forward solutions tend to oscillate on the scale of the discretization
3. The accuracy required of the adjoint solution, which is being used only for error estimation, is orders of magnitude less than generally desired for the forward solution.

An enormous literature on adaptive methods for differential equations has devel-

oped over nearly six decades of activity and the major developments form a highly inter-connected web. We do not attempt to review the history of adaptive methods or to present a comprehensive list of references. Instead, we provide only a short list of references that either contain further referenees and/or address eomputational issues related to adaptive mesh refinement for evolution problems [8, 7, 5, 4, 18, 22, 9, 10, 17, 19, 15, 3, 1, 23, 24, 20, 2, 14].

This paper considers adaptive mesh refinement from a different point of view than much of the existing literature. Namely, we are concerned with trying to understand how to adapt discretizations based on under-resolved solutions on relatively eoarse discretizations in order to obtain particular information, as opposed to analyzing adaptive mesh algorithms in the asymptotic limit of mesh refinement. This point of view is important for many large scale applications, for which sueh eonditions are generic. In §2 we review the standard *a posteriori* error analysis and modify this for a block adaptive strategy. We review adaptive error control in §3, and introduce new features necessary for block adaptivity and several block adaptive strategies. One- and three-dimensional illustrative computational examples are provided in §4 and we draw conclusions in §5.

2. Discretization and error estimation. We begin by reviewing discretization and *a posteriori* error estimation for evolution problems and then describe the block-wise discretization and present the corresponding error estimate.

2.1. Discretization. We formulate the discretization as a space-time finite element method because that is convenient for deriving *a posteriori* error estimates based on variational analysis. However, we emphasize that the estimates can be extended to a wide range of discretizations, e.g. finite difference and finite volume methods, which can be written as equivalent finite element methods.

We describe two finite element space-time discretizations of (1.1) called the continuous and discontinuous Galerkin methods, see [11, 13, 12, 10, 17, 15]. We partition $[0,T]$ as $0 = t_0 < t_1 < t_2 < \cdots < t_n < \cdots < t_N = T$, denoting each time interval by $I_n = (t_{n-1}, t_n]$ and time step by $k_n = t_n - t_{n-1}$ and we construct a discretization $\mathcal{T}$ of Ω such that the union of the elements in $\mathcal{T}$ is Ω while the intersection of any two elements is either a common edge, node, or is empty. We assume that the smallest angle of any element is bounded below by a fixed constant. To measure the size of the elements of $\mathcal{T}$, we use a piecewise constant function h, the so-called mesh function, defined so $h|_\Delta = \operatorname{diam}(\Delta)$ for $\Delta \in \mathcal{T}$. Similarly, we use k to denote the piecewise constant function that is k_n on I_n.

The approximations are polynomials in time and piecewise polynomials in space on each spaee-time "slab" $S_n = \Omega \times I_n$. In space, we let $V \subset H_0^1(\Omega)$ denote the space of piecewise linear continuous functions defined on $\mathcal{T}$, where each function is zero on $\partial\Omega$. Then on each slab, we define

$$W_n^q = \left\{ w(x,t) : w(x,t) = \sum_{j=0}^{q} t^j v_j(x),\, v_j \in V,\, (x,t) \in S_n \right\}.$$

Finally, we let W^q denote the space of functions defined on the space-time domain $\Omega \times [0,T]$ such that $v|_{S_n} \in W_n^q$ for $n \geq 1$. Note that functions in W^q may be discontinuous across the discrete time levels and we denote the jump across t_n by $[w]_n = w_n^+ - w_n^-$ where $w_n^\pm = \lim_{s \to t_n \pm} w(s)$.

We use a projection operator into V, $Pv \in V$, e.g. the L^2 projection satisfying $(Pv, w) = (v, w)$ for all $w \in V$, where $(\cdot,\cdot)$ denotes the $L_2(\Omega)$ inner product. We

use the $\|\ \|$ for the L_2 norm. We also use a projection operator into the piecewise polynomial functions in time, denoted by $\pi_n : L^2(I_n) \to \mathcal{P}^q(I_n)$, where $\mathcal{P}^q(I_n)$ is the space of polynomials of degree q or less defined on I_n. The global projection operator π is defined by setting $\pi = \pi_n$ on S_n.

DEFINITION 2.1. *The* **discontinuous Galerkin** *dG(q) approximation $U \in W^q$ satisfies $U_0^- = Pu_0$ and*

$$\int_{t_{n-1}}^{t_n} \left((\dot{U}, v) + (\epsilon \nabla U, \nabla v)\right) dt + \left([U]_{n-1}, v^+\right) = \int_{t_{n-1}}^{t_n} (f(U), v)\, dt$$
$$\text{for all } v \in W_n^q, \quad 1 \le n \le N. \quad (2.1)$$

We also use a related method for solving the adjoint problem:

DEFINITION 2.2. *The* **continuous Galerkin** *cG(q) approximation $U \in W^q$ satisfies $U_0^- = Pu_0$ and*

$$\begin{cases} \displaystyle\int_{t_{n-1}}^{t_n} \left((\dot{U}, v) + (\epsilon \nabla U, \nabla v)\right) dt = \int_{t_{n-1}}^{t_n} (f(U), v)\, dt \\ \qquad\qquad\qquad\qquad\qquad\qquad \text{for all } v \in W_n^{q-1}, \quad 1 \le n \le N, \\ U_{n-1}^+ = U_{n-1}^-. \end{cases} \quad (2.2)$$

Note that U is continuous across time nodes when the space mesh is fixed.

With appropriate use of quadrature to evaluate the integrals in the variational formulation, these Galerkin methods yield standard difference schemes. If the lumped mass quadrature is used in space, then the discrete system yielding the dG(0) approximation is the same as the system obtained for the nodal values of the "backward Euler in time"-"second order centered difference scheme in space" finite difference scheme. Likewise, the cG(1) method is related to the Crank-Nicolson scheme, and the dG(1) method is related to the third order sub-diagonal Padé difference scheme. Under general assumptions, the cG(q) and dG(q) have order of accuracy $q + 1$ in time at any point and a superconvergence order of $2q + 1$ and $2q$ respectively at time nodes.

2.2. An *a posteriori* error estimate. We begin by defining a suitable adjoint problem for error analysis. A more detailed description is given in [15]. The adjoint problem is a parabolic problem with coefficients obtained by linearization around an average of the true and approximate solutions.

$$\bar{f} = \bar{f}(u, U) = \int_0^1 \frac{\partial f}{\partial u}(us + U(1 - s))\, ds. \quad (2.3)$$

The regularity of u and U typically imply that $\bar{f}$ is piecewise continuous with respect to t and a continuous, H^1 function in space.

Written out pointwise for convenience, the adjoint problem to (1.1) for the generalized Green's function associated to the data ψ, which determines the quantity of interest,

$$\int_0^T (u, \psi)\, dt,$$

is

$$\begin{cases} -\dot{\phi} - \nabla \cdot (\epsilon \nabla \phi) - \bar{f}\phi = \psi, & (x,t) \in \Omega \times (T,0], \\ \phi(x,t) = 0, & (x,t) \in \partial\Omega \times (T,0], \\ \phi(x,T) = 0, & x \in \Omega, \end{cases} \tag{2.4}$$

This choice for the adjoint yields the following error representation formula for the dG method.

THEOREM 2.3. *dG A Posteriori Error Estimate*

$$\int_0^T (e,\psi)\,dt = ((I-P)u_0, \phi(0)) + \sum_{n=1}^{N} \left([U]_{n-1}, (\pi P\phi - \phi)^+_{n-1}\right) + \int_0^T \left((\dot{U}, \pi P\phi - \phi) + (\epsilon(U)\nabla U, \nabla(\pi P\phi - \phi)) - (f(U), \pi P\phi - \phi)\right) dt. \tag{2.5}$$

The initial error is $e^-(0) = (I-P)u_0$.

In practice, we compute a numerical solution of the linear adjoint problem obtained from (2.4) by replacing u with the computed approximate solution U in the definition of $\bar{f}$ and solve using a higher order method in space and time, see [15]. We denote the approximate adjoint solution by Φ. We focus on the dG method, while application to the cG method is analogous.

COROLLARY 2.4. *The approximate a posteriori error estimate for the dG method is*

$$\left|\int_0^T (e,\psi)\,dt\right| \approx E(U) = E(U;\psi) = \Big|((I-P)u_0, \Phi(0)) + \sum_{n=1}^{N} \left([U]_{n-1}, (\pi P\Phi - \Phi)^+_{n-1}\right) + \int_0^T \left((\dot{U}, \pi P\Phi - \Phi) + (\epsilon(U)\nabla U, \nabla(\pi P\Phi - \Phi)) - (f(U), \pi P\Phi - \Phi)\right) dt\Big|. \tag{2.6}$$

2.3. Blockwise discretization. We describe the blockwise formulation of the discontinuous Galerkin method. We partition $[0,T]$ into time blocks $0 = T_0 < T_1 < T_2 < \cdots < T_b < \cdots < T_B = T$. We discretize each block $[T_{b-1}, T_b]$ by $T_{b-1} = t_{b,0} < t_{b,1} < \cdots < t_{b,N_b} = T_b$, denoting each subinterval by $I_{b,n} = (t_{b,n-1}, t_{b,n}]$ and time step by $k_{b,n} = t_{b,n} - t_{b,n-1}$. To each block $[T_{b-1}, T_b]$, we associate a discretization $\mathcal{T}_b$ of Ω arranged so the union of the elements in $\mathcal{T}_b$ is Ω while the intersection of any two elements is either a common edge, node, or is empty. We assume that the smallest angle of any element is bounded below by a fixed constant. To measure the size of the elements of $\mathcal{T}_b$, we use the mesh function h_b.

The approximations are polynomials in time and piecewise polynomials in space on each space-time "slab" $S_{b,n} = \Omega \times I_{b,n}$. In space, we let $V_b \subset H_0^1(\Omega)$ denote the space of piecewise linear continuous functions defined on $\mathcal{T}_b$, where each function is zero on $\partial\Omega$. Then on each slab, we define

$$W^q_{b,n} = \left\{ w(x,t) : w(x,t) = \sum_{j=0}^{q} t^j v_{b,j}(x),\ v_{b,j} \in V_b,\ (x,t) \in S_{b,n} \right\}.$$

Finally, we let W^q denote the space of functions defined on the space-time domain $\Omega \times [0,T]$ such that $v|_{S_{b,n}} \in W^q_{b,n}$ for $b,n \geq 1$. Note that functions in W^q may be

discontinuous across the discrete time levels and we denote the jump across $t_{b,n}$ by $[w]_{b,n} = w^+_{b,n} - w^-_{b,n}$.

To compute the dG approximation on the new block, we project the final value of the approximation from the previous block onto the new mesh. We use a projection operator $P_b v \in V_b$ and a projection operator into the piecewise polynomial functions in time, denoted by $\pi_{b,n} : L^2(I_{b,n}) \to \mathcal{P}^q(I_{b,n})$. We then define π_b as $\pi_b = \pi_{b,n}$ on $S_{b,n}$. Finally, we define global projections P and π which on each block are P_b and π_b respectively.

DEFINITION 2.5. *The blockwise discontinuous Galerkin dG(q) approximation $U \in W^q$ satisfies $U^-_{b,0} = P_1 u_0$ and for $b = 1, 2, \cdots, B$,*

$$\int_{t_{b,n-1}}^{t_{b,n}} \left((\dot{U}, v) + (\epsilon \nabla U, \nabla v)\right) dt + \left([U]_{b,n-1}, v^+\right) = \int_{t_{b,n-1}}^{t_{b,n}} (f(U), v)\, dt$$
$$\text{for all } v \in W^q_{b,n}, \quad 1 \le n \le N_b. \quad (2.7)$$

2.4. A blockwise *a posteriori* error estimate. Adapting the standard argument that yields (2.5), we obtain a blockwise *a posteriori* error estimate.

THEOREM 2.6. *Blockwise A Posteriori Error Estimate*

$$\int_0^T (e, \psi)\, dt \approx ((I - P_0)u_0, \Phi(0)) + \sum_{b=1}^{B} \left((I - P_b)U, \Phi(T_{b-1})\right)$$
$$+ \sum_{b=1}^{B} \Bigg(\int_{T_{b-1}}^{T_b} \left((\dot{U}, \pi P_b \Phi - \Phi) + (\epsilon(U)\nabla U, \nabla(\pi P_b \Phi - \Phi)) - (f(U), \pi P_b \Phi - \Phi)\right) dt$$
$$+ \sum_{n=1}^{N_b} \left([U]_{b,n-1}, (\pi P_b \Phi - \Phi)^+_{b,n-1}\right) \Bigg). \quad (2.8)$$

The second term on the right measures the effects of changing meshes on the accuracy of the approximation. A similar "jump" term already appears in the estimate for the standard dG method at each time step. In this case of transitions between blocks, the "jump" arises because of mesh changes between blocks. Note that the adjoint weight does not involve the projection of Φ into the approximation space (i.e. Galerkin orthogonality). Instead, the contributions from the projections accumulate in the same way as an initial error.

Our purpose is to use the *a posteriori* bounds $\text{Æ}_x, \text{Æ}_t$ to choose block times $\{T_b\}$ and corresponding meshes $\mathcal{T}_b$ and timesteps $k_{b,i}$. An important issue is the effect of transferring solutions between the meshes of adjacent blocks on the accuracy of the computed information, and so we address the computation of a bound on the second term on the right in (2.8),

$$\Xi(U) = \sum_{b=1}^{B} \left|\left((I - P_b)U, \Phi(T_{b-1})\right)\right|. \quad (2.9)$$

3. Adaptive error control. We start off by describing some standard approaches to adaptive error control and the relation to adaptive error control based on *a posteriori* error estimates. We then turn to the problem of choosing blocks for a block discretization and generating the corresponding spatial and temporal discretizations for each block.

3.1. Goal oriented adaptive error control. The aim of goal oriented adaptive error control is to generate a mesh with a nearly minimal number of elements such that for a given tolerance TOL and data ψ,

$$\left| \int_0^T (e, \psi)\, ds \right| \lesssim \text{TOL}. \tag{3.1}$$

We note that (3.1) cannot be verified in practice because the error is unknown, so instead we use an estimate or a bound for the error in the quantity of interest. Different ways to generate an acceptable mesh vary by the estimate or bound used for the quantity of interest as well as the strategy for mesh refinement.

For example using the *a posteriori* estimate (2.6), the goal of adaptive error control is to determine a discretization so that a mesh acceptance criterion,

$$E(U) \lesssim \text{TOL}, \tag{3.2}$$

is satisfied. If (3.2) is not satisfied, then we refine the mesh in order to compute a new solution for which the criterion is met. Refinement decisions require identifying the contributions to the error from discretization on each element. We can write $E(U)$ as a sum over space-time elements,

$$E(U) = \Big| \sum_{\Delta \in \mathcal{T}} ((I - P)u_0, \Phi(0))_\Delta + \sum_{n=1}^{N} \sum_{\Delta \in \mathcal{T}} \left([U]_{n-1}, (\pi P \Phi - \Phi)^+_{n-1} \right)_\Delta$$
$$+ \sum_{n=1}^{N} \sum_{\Delta \in \mathcal{T}} \int_{t_{n-1}}^{t_n} \left((\dot{U}, \pi P \Phi - \Phi)_\Delta + (\epsilon(U) \nabla U, \nabla(\pi P \Phi - \Phi))_\Delta - (f(U), \pi P \Phi - \Phi)_\Delta \right) dt \Big|,$$

where $(\ ,\ )_\Delta$ denotes the L^2 inner product on element Δ. This clearly identifies possible element contributions.

However, a major difficulty is that the error estimate generally involves a large amount of cancelation among the element contributions, which makes determining a truly efficient refinement strategy extremely difficult.

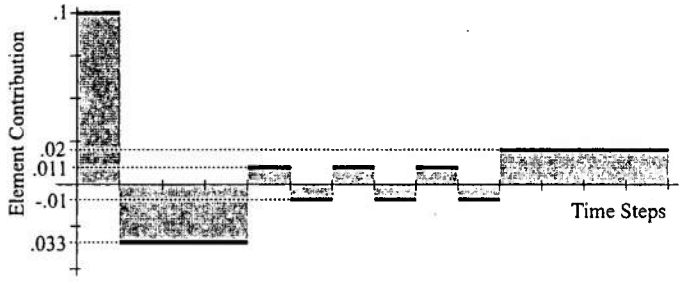


FIG. 3.1. *The element contributions to the error in integration.*

EXAMPLE 3.1. We consider a first order accurate numerical solution that has the element contributions shown in Fig. 3.1.

The first time step has the largest contribution. The next three steps each contribute -0.033, so cancelation means that the total contribution from the first four

steps is 0.001. Likewise, the next six steps contribute +0.003 in total. The last four steps contribute 0.08 in total. The total error is therefore

$$.1 - 3 \times .033 + .011 - .01 + .011 - .01 + .011 - .01 + 4 \times .02 = 0.084$$

If we use a standard approach of refining only some fraction of the elements with the largest contributions, we are likely to refine the first four steps. For simplicity, we assume that the elements marked for refinement are divided into two time steps. The resulting integration will have accuracy

$$\frac{1}{2^2} \times 2 \times .1 - \frac{1}{2^2} \times 6 \times .033 + .011 - .01 + .011 - .01 + .011 - .01 + 4 \times .02 \approx 0.0835.$$

Note that the individual element contributions decrease at a second order rate. The problem is that even though the element contributions in the first four steps are individually large, there is significant cancelation and refinement in this region and refinement does not decrease the error significantly. On the other hand, if we refine the last four time steps instead, we obtain

$$.1 - 3 \times .033 + .011 - .01 + .011 - .01 + .011 - .01 + \frac{1}{2^2} \times 8 \times .02 \approx 0.044.$$

While this is a non-standard approach, it decreases the error significantly.

In the adjoint-weight approach, the issue of cancelation of error is neglected in a sense by replacing the *accurate* error estimate $E(U)$ by an inaccurate upper bound,

$$E(U) \leq \text{Æ}(U) = \text{Æ}(U;\psi), \tag{3.3}$$

where we define $\text{Æ}(U;\psi)$ by summing bounds over each element.

DEFINITION 3.2. *Element-wise upper bound on the total error*

$$\text{Æ}(U;\psi) = \sum_{\Delta\in\mathcal{T}} |((I-P)u_0, \Phi(0))_\Delta| + \sum_{n=1}^{N}\sum_{\Delta\in\mathcal{T}} \left| \left([U]_{n-1}, (\pi P\Phi - \Phi)^+_{n-1}\right)_\Delta \right|$$

$$+\sum_{n=1}^{N}\sum_{\Delta\in\mathcal{T}} \left| \int_{t_{n-1}}^{t_n} (\dot{U}, \pi P\Phi - \Phi)_\Delta + (\epsilon(U)\nabla U, \nabla(\pi P\Phi - \Phi))_\Delta - (f(U), \pi P\Phi - \Phi)_\Delta \, dt \right|.$$

Thus, if (3.2) is not satisfied, the mesh is refined in order to achieve

$$\text{Æ}(U) \lessapprox \text{TOL}. \tag{3.4}$$

The adaptive error control problem can now be profitably posed as a constrained minimization problem, namely to find a mesh with a minimal number of degrees of freedom on which the approximation satisfies the bound (3.4). Using the fact that the bound Æ is a sum of positive terms and assuming the solution is asymptotically accurate, a calculus of variations argument yields the generic (see e.g. [9, 10, 3, 2]).

> ***Principle of Equidistribution*** An approximate solution of the constrained optimization problem for an optimal mesh for an upper bound on the error is achieved when the elements contributions to the bound are approximately equal.

The Principle of Equidistribution has been used in various forms at least since the seventies (and probably earlier in industry). However, experience with a wide range of problems suggest that the bound $Æ(U)$ is generically several orders of magnitude larger than the estimate $E(U)$. A strategy based on the Principle of Equidistribution that optimizes computational cost with respect to a error *bound* and not the actual error can therefore result in significant over-refinement.

In general, there are many solutions of the constrained minimization problem associated with (3.4). An adaptive mesh algorithm is a procedure for computing an acceptable solution. Traditionally, different approaches are used for spatial and temporal adaption. A global "compute-estimate-mark-adapt" algorithm (see for example 1.1) is typically used for spatial meshes. This is an iterative approach in which only some fraction of the elements on which the contribution to the error bound is largest are refined during each iteration and whole cycle is iterated until a prescribed tolerance is achieved. Temporal approaches to mesh adaption, e.g., local error control [21], tend to use a "sweeping" strategy from initial to final time, where a solution is advanced past each time step only when the step contribution is estimated to be lower than an acceptable fraction of the total error. This may be viewed as a generally pessimistic way to achieve the Principle of Equidistribution because it removes positive effects of cancelation of error altogether. As a consequence of these differences, element contributions to the error estimate or bound typically vary in size quite considerably while contributions from different time intervals are more nearly equal.

We use a strategy that treats space and time discretizations more equitably. In the case of a parabolic problem, it is straightforward to distinguish the time and space contributions to the bound $Æ$. We define the time and space bounds as follows.

DEFINITION 3.3. *Element-wise temporal and spatial error bounds*

$$\begin{aligned} Æ_t(U) = &\sum_{n=1}^{N} \sum_{\Delta \in \mathcal{T}} \left| \left([U]_{n-1}, ((\pi - I)P\Phi)^+_{n-1} \right)_\Delta \right| \\ &+ \sum_{n=1}^{N} \sum_{\Delta \in \mathcal{T}} \left| \int_{t_{n-1}}^{t_n} (\dot{U}, (\pi - I)P\Phi)_\Delta + (\epsilon(U)\nabla U, \nabla(\pi - I)P\Phi)_\Delta \right. \\ &\qquad \left. - (f(U), (\pi - I)P\Phi)_\Delta \, dt \right|, \end{aligned} \tag{3.5}$$

$$\begin{aligned} Æ_x(U) = &\sum_{\Delta \in \mathcal{T}} \left| ((I - P)u_0, \Phi(0))_\Delta \right| + \sum_{n=1}^{N} \sum_{\Delta \in \mathcal{T}} n \left| \left([U]_{n-1}, (P\Phi - \Phi)^+_{n-1} \right)_\Delta \right| \\ &+ \sum_{n=1}^{N} \sum_{\Delta \in \mathcal{T}} \left| \int_{t_{n-1}}^{t_n} (\dot{U}, P\Phi - \Phi)_\Delta + (\epsilon(U)\nabla U, \nabla(P\Phi - \Phi))_\Delta \right. \\ &\qquad \left. - (f(U), P\Phi - \Phi)_\Delta \, dt \right|. \end{aligned} \tag{3.6}$$

We see that the time bound is precisely the *a posteriori* bound for the dG approximation for the "method of lines" initial value problem resulting after discretization in space. The adjoint weight depends on the projection of the adjoint solution into the time finite element space. On the other hand, the adjoint weight in the space bound depends on the projection of the adjoint solution into the spatial finite element space.

We split the error between the time and space contributions and refine the current mesh in order to achieve

$$Æ_x(U) \lessapprox \frac{\mathrm{TOL}}{2} \text{ and } Æ_t(U) \lessapprox \frac{\mathrm{TOL}}{2}. \tag{3.7}$$

On a given time interval, this requires an iteration during which both the space mesh and time steps are refined.

3.2. Goal oriented block adaptive error control. For the purpose of developing a block adaptive algorithm, we treat adaptivity with respect to space and time in the same way. The reason is that we determine the blocks by predicting the local element sizes (or number of sub-elements) that are required in the final mesh. We create a block by grouping together a set of coarse-scale space-time slabs that are adjacent in time and satisfy some criteria, e.g. similar spatial meshes are predicted for the space-time slabs in the block or a maximal number of elements are predicted to be required in the block.

3.2.1. Choosing a global tolerance for the error bound. We want the predictions of the element sizes required in an acceptable fine scale mesh to be as accurate as possible. We recall that an acceptable mesh need only satisfy the estimate criterion (3.2) and not the more stringent bound criterion (3.4). We define the overestimation factor for a given mesh,

$$\gamma = \frac{Æ(U)}{E(U)},$$

and the corresponding absolute tolerance for Æ,

$$\mathrm{ATOL} = \gamma \times \mathrm{TOL}.$$

We replace (3.4) by

$$Æ_x(U) \lessapprox \frac{\mathrm{ATOL}}{2} \text{ and } Æ_t(U) \lessapprox \frac{\mathrm{ATOL}}{2}. \tag{3.8}$$

Note that ATOL $\approx$ TOL when there is little cancelation among the element contributions and ATOL $>$ TOL otherwise. In this way, we attempt to mitigate the inefficiency that is introduced by replacing an accurate error estimate by an inaccurate bound in decisions about mesh refinement. This approach for setting tolerances is discussed further in [16].

3.2.2. Predicting refinement in space. Given a local space-time element $\mathfrak{S} = \mathfrak{S}(\triangle, n) = \triangle \times [t_{n-1}, t_n]$ in the n^{th} space-time slab that is marked for refinement, we show how to predict the number of space-time elements that are needed to meet the acceptance criterion. We assume that in the current mesh, there are N time steps and M space elements in each space-time slab, giving a total of NM space-time elements. We define a local absolute tolerance

$$\mathrm{LATOL} = \frac{\mathrm{ATOL}}{2NM}.$$

By the Principle of Equidistribution, we adopt the goal of refining each space-time element so that the local element contribution is approximately LATOL.

Using *a priori* convergence analysis, see [15], it is possible to show that there is a constant C such that

$$Æ_x\big|_{\mathfrak{S}(\Delta,n)} \sim C(h_\Delta)^p \tag{3.9}$$

as $h_\Delta \to 0$, where p is related to the order of the finite element method in space and h_Δ is the element size. Likewise, we can show constant C such that

$$Æ_t\big|_{\mathfrak{S}(\Delta,n)} \sim Ck^q \tag{3.10}$$

as $k \to 0$, where q is related to the order of the finite element method in time.

Now suppose that an element $\mathfrak{S}_{\text{new}}$ in the final mesh is obtained from $\mathfrak{S}_{\text{old}}$ in the current mesh by refinement. We have

$$\text{LATOL} \approx Æ_x\big|_{\mathfrak{S}_{\text{new}}} \approx Æ_x\big|_{\mathfrak{S}_{\text{old}}} \times \left(\frac{h_{\Delta_{\text{new}}}}{h_{\Delta_{\text{old}}}}\right)^p. \tag{3.11}$$

This yields a prediction for the new mesh size,

$$h_{\Delta_{\text{new}}} \approx \left(\frac{\text{LATOL}}{Æ_x\big|_{\mathfrak{S}_{\text{old}}}}\right)^{1/p} \times h_{\Delta_{\text{old}}}. \tag{3.12}$$

Recalling that d is the space dimension, this predicts that the element Δ_{old} should be refined into roughly

$$\left(\frac{h_{\Delta_{\text{old}}}}{h_{\Delta_{\text{new}}}}\right)^d = \left(\frac{Æ_x\big|_{\mathfrak{S}_{\text{old}}}}{\text{LATOL}}\right)^{d/p} \tag{3.13}$$

sub-elements.

3.2.3. Predicting refinement in time. For refinement in time,

$$Æ_t\big|_{\mathfrak{S}_{\text{new}}} \approx Æ_t\big|_{\mathfrak{S}_{\text{old}}} \times \left(\frac{k_{\text{new}}}{k_{\text{old}}}\right)^q \approx \text{LATOL}. \tag{3.14}$$

This yields a prediction for the new mesh size,

$$k_{\text{new}} \approx \left(\frac{\text{LATOL}}{Æ_t\big|_{\mathfrak{S}_{\text{old}}}}\right)^{1/q} \times k_{\text{old}}. \tag{3.15}$$

This predicts that the time step k_{old} should be refined into roughly

$$\frac{k_{\text{old}}}{k_{\text{new}}} = \left(\frac{Æ_t\big|_{\mathfrak{S}_{\text{old}}}}{\text{LATOL}}\right)^{1/q} \tag{3.16}$$

sub-intervals.

3.2.4. Determining overlaps for meshes on adjacent blocks. After the meshes for each block are determined based on the *a posteriori* prediction of error, we need to estimate the effects of transferring the solution between meshes on adjacent blocks. See § 4.1 for an example that illustrates this point. Recall that (2.9) provides a bound on these effects. The difficulty with using (2.9) is that we do not have the fine scale numerical solution U required for that expression until after solving on the fine scale, whereas ideally we could predict a reasonable overlap before computing the expensive fine scale solution.

We list three strategies for mitigating the possibility of projection error in our block adaptive framework.

1. There is a very simple strategy. In forming the space mesh for the block $[T_{b-1}, T_b] \times \Omega$, we guide refinement by using the maximum of the element contributions on each individual element, taking the maximum over the time intervals included in the block. We may simply include the maximum over the last time interval included in the previous block, $[T_{b-2}, T_{b-1}]$, i.e., over the interval $[t_{b-1,N_{b-1}-1}, t_{b-1,N_{b-1}}]$. We can be even more conservative by including some number of the last time steps in the maximum computation.
2. We can use gradient recovery [6] to compute an approximate solution on the fine scale mesh in each block using the solution from the last time interval contained in each block. We can then directly compute $(I - P_b)U$ for each b and evaluate (2.9).
3. We can evaluate (2.9) *a posteriori* by evaluating $(I - P_b)U$ using the fine scale forward solution and the coarse scale adjoint solution.

3.3. Block adaptive algorithms. Using the development above, we present a generic block adaptive algorithm in Algorithm 3.1. We provide a detailed algorithm in Appendix A.

Algorithm 3.1 Block Adaptive Algorithm

1: Choose the "coarse" mesh and time step
2: Compute the coarse scale numerical solution
3: Estimate the element contributions to the error for the current solution
4: Predict the number of space-time elements into which each current space-time element is to be divided using (3.13) and (3.16)
5: Build block discretizations by constructing meshes satisfying the requirements for groups of neighboring time steps
6: Compute the fine scale numerical solution using the block discretizations

We note that the Block Adaptive Algorithm 3.1 can be iterated, so that the fine scale becomes the new coarse scale, and a new fine scale is subsequently computed. In crude terms, the block adaptive Algorithm 3.1 is analogous to the core estimate-mark-refine algorithm at the heart of the generic Algorithm 1.1, but is different in the mark and refine steps. The critical step defining the block adaptive algorithm Algorithm 3.1 is the strategy used to create block discretizations. Once the blocks are identified, we can use any adaptive mesh refinement strategy for producing the actual meshes. We describe several strategies for determining block discretizations.

3.3.1. A memory-bound strategy. In the first strategy, we assume there is a target number of elements `Nmax` in space that is maximal in some sense, e.g., the largest number of elements that can be stored in core. We form blocks by creating a

union of adjacent coarse-scale space-time slabs, one slab at a time, until the projected space mesh for the block uses Nmax elements. To create the block mesh, we use the maximum of the predicted number of elements Nelem_children on each individual element (given by equation (3.13)) in the union forming the block. We illustrate in Fig. 3.2. The parameter θ governs how often the mesh is replaced by a coarser mesh, where $\theta \approx 10$ works well in practice.

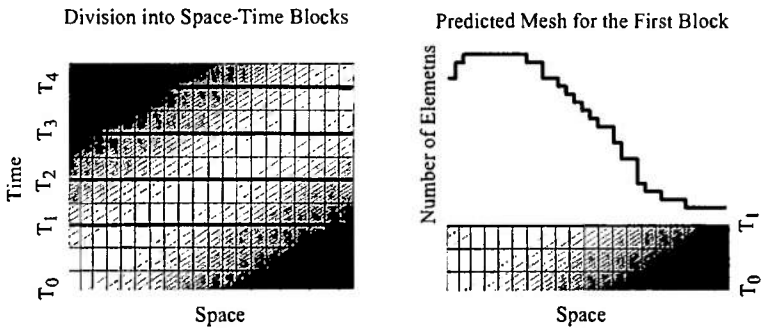


FIG. 3.2. *The memory bound strategy is used for a traveling pulse that moves with constant speed from left to right. Left: The original uniform mesh and a contour plot of the number of predicted elements of new sub-elements* Nelem_children. *The scale is from dark (low) to white (high). Right: The predicted number of new sub-elements* Nelem_children *for the first block, which consists of three adjacent space-time slabs from the original discretization.*

3.3.2. A correlation strategy. In the second strategy, we aim to choose blocks in order to use a relatively small number of elements, so Nmax may be considerably smaller than for the first algorithm. This strategy forms a block by grouping together adjacent coarse-scale space-time slabs whose predicted number of elements Nelem_children are close.

In [14], we consider the problem of detecting significant overlap of local element contributions for different computations. Following the approach there, given two vectors $\vec{v}, \vec{w}$ whose coefficients are element contributions to an error estimate, we define their *correlation* to be $c(\vec{v}, \vec{w}) = \vec{v} \cdot \vec{w}$. We say that $\vec{v}$ is significantly correlated with $\vec{w}$ if

$$\frac{c(\vec{v}, \vec{w})}{\|\vec{w}\|^2} > \gamma_1 \text{ and } \frac{\|\vec{w} - \frac{c(\vec{v}, \vec{w})}{\|\vec{v}\|^2}\vec{v}\|}{\|\vec{w}\|} < \gamma_2.$$

where $0 < \gamma_1, \gamma_2$. The first condition insures that $\vec{v}$ has a suitable large projection onto $\vec{w}$ while the second condition corrects for differences in scale between $\vec{v}$ and $\vec{w}$ (consider $\|\vec{v}\| \gg \|\vec{w}\|$ so that $c(\vec{v}, \vec{w}) \gg \|\vec{w}\|$).

We implement the new criterion for creating blocks by choosing to add the next time slab to a current block based on the correlation criterion.

3.3.3. Global strategies. In the first two strategies for creating blocks, we sweep through time. We can also use a bisection search beginning with the original large block and subdividing to find acceptable blocks. In analog to the difference between the standard global strategy for space mesh refinement to achieve the Principle of Equidistribution and the local-error control approach, the bisection search is a global strategy that can be a more efficient way to achieve equidistribution.

4. Computational Examples. We apply the block adaptive algorithms to several prototypical examples in one and three space dimensions. The one dimensional examples illustrate several key points when implementing block-adaptive methods, while the three dimensional examples include a traveling wave front, a solution that undergoes time- and space-localized perturbations, and a periodic motion in a convection-dominated flow.

The forward problems and adjoint problems are solved with linear and quadratic elements in space and dG0 and cG1 in time respectively. The one dimensional examples are computed using the Matlab code ACES [25]. The three dimensional examples are performed on a hexahedral mesh using a trilinear spatial basis for the forward problem and a triquadratic basis for the adjoint. Local mesh refinement is accomplished by the use of hanging nodes where one hanging node per edge or face is allowed. Conformity of the basis is obtained by interpolation of the surrounding regular nodes. The use of an hierarchical octree-based data structure assists refinement but also allows for de-refinement when the element indicators are small. For the convection driven flow problem, SUPG is employed for both the forward and adjoint problems, with parameter

$$\delta = \frac{1}{(1/\Delta t + U/h)},$$

where Δt is the time step and U is the speed of the convection field at the current time, i.e., $U = ||\beta||_2$ in (4.5). This is not an obstacle for the block-adaptive framework, as we simply modify the theoretical convergence rate p in the computation of `Nelem_children` in (3.13).

4.1. Example One: Projection errors between blocks. We illustrate the necessity for addressing the effect of transferring solutions between space-time blocks with a simple one-dimensional example involving a traveling wave.

$$\begin{cases} u_t - u_{xx} = f(x,t), & 0 < x < 1,\ 0 < t, \\ u(0,t) = u(1,t) = \beta(t), & 0 < t, \\ u(x,0) = \tanh(\alpha(x-0.2)), & 0 < x < 1, \end{cases} \tag{4.1}$$

where $\alpha = 50$ and f and β are chosen to give an exact solution $u = \tanh(\alpha(x-t-0.2))$. We solve with a coarse mesh using $h = 0.1$ and time step $k = 0.05$ from initial time 0 to final time 0.6. The quantity of interest is the average space-time error. We compute a fine scale solution using two blocks derived from the coarse scale solution. The first block, $t = [0, 0.3]$, uses a finer spatial mesh in the region $x \in [0.1, 0.6]$, while the second block uses a fine mesh in the region $[0.5, 1]$, so the overlap is minimal and and the predictions for refinement areas are incorrect. Consequently, the approximate traveling wave travels too quickly. The first block solution at $t = 0.3$ and its projection onto the second block at $t = 0.3$ is displayed in Fig. 4.1.

In Fig. 4.1 we illustrate the *a posteriori* use of (2.9) to correct the projection error. Block 1 is computed using the predicted fine scale mesh. Block 2 is tested for significant projection error using (2.9) using the fine scale solution for Block 1 and the mesh for Block 2 is refined if the elementwise projection error exceeds *LATOL*. We note that the overlap strategy for the projection error in §3.2.4 also works well in this particular example.

4.2. Example Two: Coarse scale resolution. Since we are using the coarse scale discretization to predict the global behavior of the solution on the fine scale,

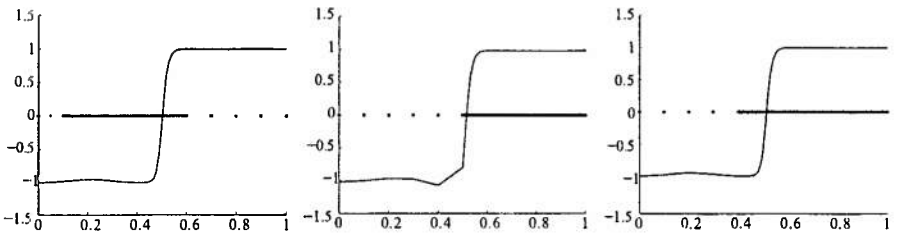


FIG. 4.1. *Problem (4.1). The circles indicate the spatial meshes used in each of the two blocks. Left: the solution on Block 1. Middle: the projection of the approximate solution in block 1 onto the mesh in block 2. Right: the solution onto Block 2 after using the projection error estimate (2.9) to correct significant projection errors between the two blocks. This demonstrates the possible consequences when the meshes for neighboring blocks do not overlap sufficiently.*

it is important to insure that the coarse scale discretization is not too coarse. (This is a difference between the block adaptive approach and a standard adaptive mesh refinement, which is generally started with a very coarse mesh.) This issue is especially important for nonlinear problems since linearization is used to define the adjoint problem, which in turn provides the means to quantify the effects of cancelation and accumulation of errors.

Consider the one-dimensional nonlinear parabolic equation

$$\begin{cases} u_t - \frac{1}{2\alpha} u_{xx} = \alpha(u-1)(1-u^2), & -1 < x < 1,\ 0 < t < 0.6, \\ u(0,t) = -1,\ u(1,t) = 1, & 0 < t, \\ u(x,0) = \tanh(\alpha(x-0.2)), & -1 < x < 1, \end{cases} \tag{4.2}$$

We choose α to obtain the same solution as the example in § 4.1, $u = \tanh(\alpha(x - t - 0.2))$. The quantity of interest is the average space-time error. For the coarse discretization, we use $h = 0.05$ and $k = 0.05$. These choices provide an excellent coarse scale discretization for the linear example in § 4.1 but does not work well for the nonlinear version. We show two snapshots of the solution u in Fig. 4.2 at $t = 0.3$ and $t = 0.6$. The wave-speed is predicted inaccurately, which leads to a poor block selection and this subsequently affects the fine scale accuracy. Using a coarse scale discretization with $h = 0.1$ and $k = 0.1$ yields inaccurate results.

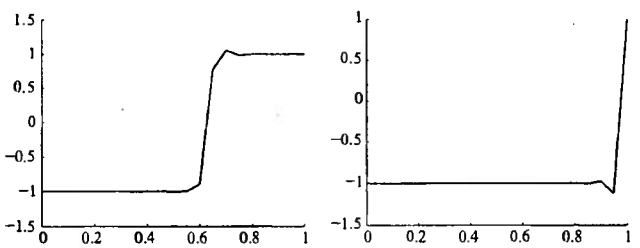


FIG. 4.2. *Problem (4.2). Correlation strategy with an insufficiently accurate coarse-scale solution. Solution on the adapted mesh at $t = 0.3$ and $t = 0.6$ respectively.*

The poor predictions based on the coarse-scale discretization can be avoided by slightly enriching the discretization with a finer time step. We use a coarse discretiza-

tion with $h = 0.05$ and $k = 0.01$ and the correlation strategy to produce blocks. The approximate solution on the adapted mesh at $t = 0.45$ is shown in Fig. 4.3.

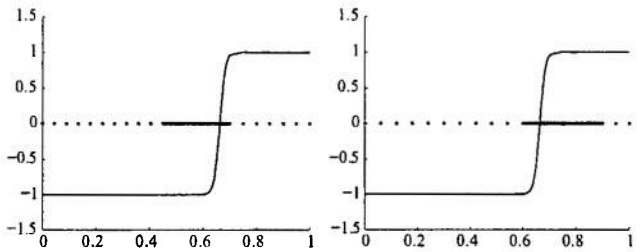


FIG. 4.3. *Problem (4.2). Correlation strategy with an improved coarse-scale solution. Solution on the adapted mesh at $t = 0.45$ on blocks 3 and 4 respectively.*

4.3. Example three: A traveling wave solution. This example is a wave propagating along the main diagonal of the unit cube ($\Omega = [0,1] \times [0,1] \times [0,1]$). The governing equation is

$$\begin{cases} u_t - \Delta u = f(x,t), & x \in \Omega,\ 0 < t, \\ u(x,t) = 0, & x \in \partial\Omega, 0 < t, \\ u(x,0) = (x_1 - x_1^2)(x_2 - x_2^2)(x_3 - x_3^2)\arctan(\frac{c\sqrt{3}}{3}\sqrt{x_1^2 + x_2^2 + x_3^2}), & x \in \Omega, \end{cases} \tag{4.3}$$

where $c = 75$ and f is constructed to yield the exact solution

$$u = \frac{\sqrt{3}}{3}\arctan(\frac{c\sqrt{3}}{3}\sqrt{x_1^2 + x_2^2 + x_3^2} - t).$$

The coarse block solution u_C is constructed on an $8 \times 8 \times 8$ uniform mesh using hexahedral meshes with an initial time step of 0.1. The quantity of interest is the time average of the solution value. The memory bound strategy is used to construct the discretization blocks with $ATOL = 0.000178$ and Nmax=50000. Block information is given in Table 4.1. As might be expected, all of the blocks use approximately the same number of elements. We show contour plots of the solution on "slices" of some of the block meshes along the plane $x = 0.5$ in Fig. 4.4.

Block	T_{b-1}	T_b	# vertices	# hexahedra
1	0	0.4	58711	50394
2	0.4	0.6	63219	54503
3	0.6	0.7	72267	61265
4	0.7	0.8	62626	52368
5	0.8	1	64764	54860
6	1	1.1	62790	54377

TABLE 4.1
Problem (4.3). Blocks resulting from the memory bound strategy.

4.4. Example Four: Localized forcing in space and time. This example contrasts the difference in the blocks produced by the memory bound and correlation

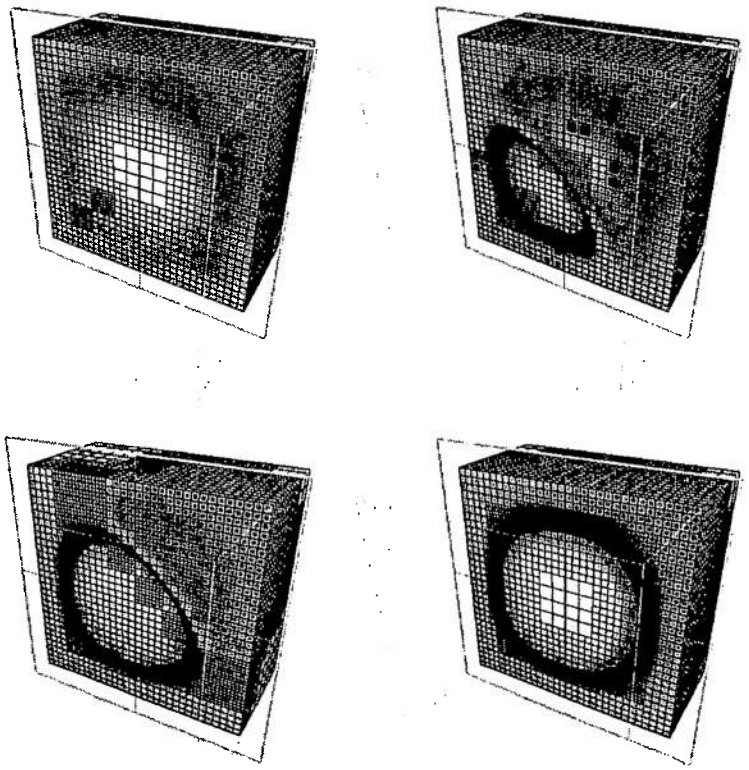


FIG. 4.4. *Problem (4.3). Memory bound strategy. Slices through the mesh perpendicular to the x-axis at $x = 0.5$. Upper left: $t = 0$ (block 1). Upper right: $t = 0.44$ (block 2). Lower left: $t = 0.6$ (block 3). Lower right: $t = 1.1$ (block 6).*

strategies when solving an equation with source terms that are localized in space and time. The governing equation on the unit cube Ω is

$$\begin{cases} u_t - \Delta u = 50e^{-(\alpha_1(x-x_1)^2+(t-t_1)^2)} + 20e^{-(\alpha_2(x-x_2)^2+(t-t_2)^2)}, & x \in \Omega, 0 < t, \\ u(x,0) = 0, & x \in \Omega, \end{cases} \tag{4.4}$$

with homogeneous Neumann boundary conditions on all the sides except the bottom where a homogeneous Dirichlet condition is imposed. We choose $\alpha_1 = 50$, $\alpha_2 = 10$, $t_1 = 1$, $t_2 = 10$, $x_1 = (0.125, 0.125, 0.125)$ and $x_2 = (0.75, 0.5, 0.75)$. The quantity of interest is the time average of the solution value.

We use a coarse discretization consisting of an $8 \times 8 \times 8$ uniform hexahedral mesh and time step of 0.1. With $ATOL = 0.00010044$ and $\texttt{Nmax} = 50000$ we show the block information for the memory bound and correlation strategies respectively in Table 4.2 and Table 4.3. The algorithms lead to significantly different block meshes.

Block	T_{b-1}	T_b	# vertices	# hexahedra
1	0	1.1	59465	54125
2	1.1	1.2	63112	57772
3	1.2	2.4	45359	40958
4	2.4	11.9	12383	10165
5	11.9	14.9	2029	1478

TABLE 4.2
Problem (4.4). Blocks resulting from the memory bound strategy.

Block	T_{b-1}	T_b	# vertices	# hexahedra
1	0	1.1	63112	57772
2	1.1	1.2	63112	57772
3	1.2	1.6	45359	40958
4	1.6	2.5	9611	8037
5	2.5	2.9	1968	1436
6	2.9	8.5	966	652
7	8.5	9	2617	1926
8	9	10.8	12651	10382
9	10.8	11.3	7363	5860
10	11.3	12.6	3139	2360
11	12.6	14.9	729	512

TABLE 4.3
Problem (4.4). Blocks resulting from the correlation strategy.

The correlation strategy chooses many more blocks, but many of the blocks have very low numbers of elements.

We show planar slices near x_1 and x_2 of the meshes for Blocks 1 and 3 in Fig. 4.5. For comparison, we show planar slices perpendicular to the x-axis near x_1 and x_2 of the meshes for blocks constructed using the two strategies in Fig. 4.6. Both strategies result in similar meshes near x_2 at time $t = 10$. However at $t = 8.8$, the correlation strategy leads to coarse meshes that are not produced by the memory bound strategy. The mesh resulting from the memory bound strategy retains the refinement resulting from the earlier perturbation near x_1 at $t = 1$.

4.5. Example Five: Periodic motion in a convection-dominated flow. This example has a heat source with a forced oscillating convective term within the unit cube Ω to produce an "orbiting" zone of perturbation. The governing equation is

$$\begin{cases} u_t + \beta \cdot \nabla u - \Delta u = f, & x \in \Omega, 0 < t < 1, \\ u(x,t) = 0, & x \in \partial\Omega, 0 < t < 1, \\ u(x,0) = 0, & x \in \Omega, \end{cases} \tag{4.5}$$

with $\beta = (20(\cos(\pi t)\sin(2\pi t), \sin(\pi t)\sin(2\pi t), \cos(2\pi t))$ and $f(x) = e^{-50(x_1^2+x_2^2+x_3^2)}$. The quantity of interest is the time average value. The coarse discretization used 4913 vertices and at time step of 0.01. The blocks constructed by the memory-bound strategy using $ATOL = 0.00044$ and Nmax=50000 are described in Table 4.4.

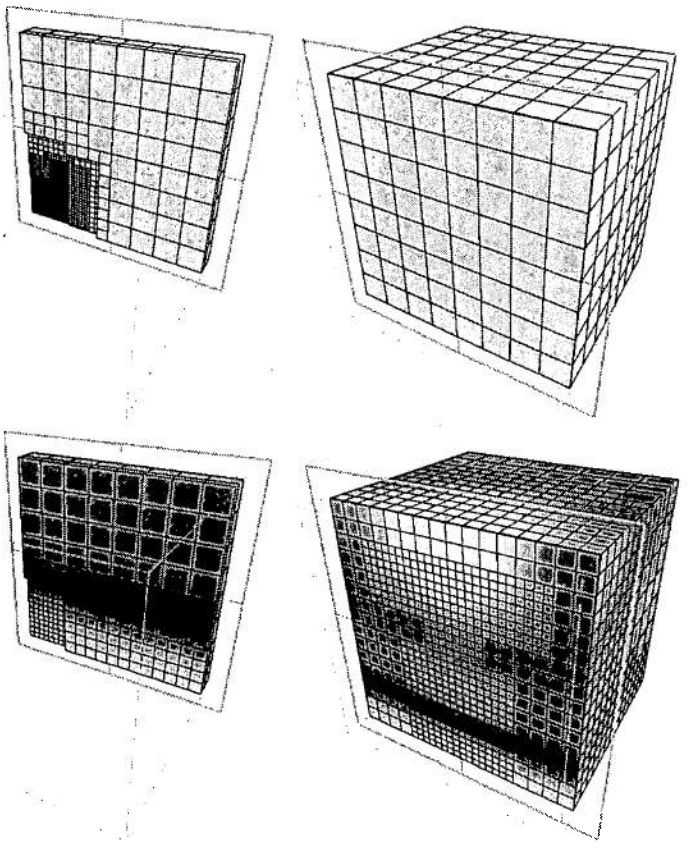


FIG. 4.5. *Problem (4.4). Memory bound strategy. Slices through the mesh perpendicular to the x-axis. Upper left: Slice near* x_1 *at* $t = 1$ *(block 1). Upper right: Slice near* x_2 *at* $t = 1$ *(block 1). Lower left: Slice near* x_1 *at* $t = 10$ *(block 4). Lower right: Slice near* x_2 *at* $t = 10$ *(block 4).*

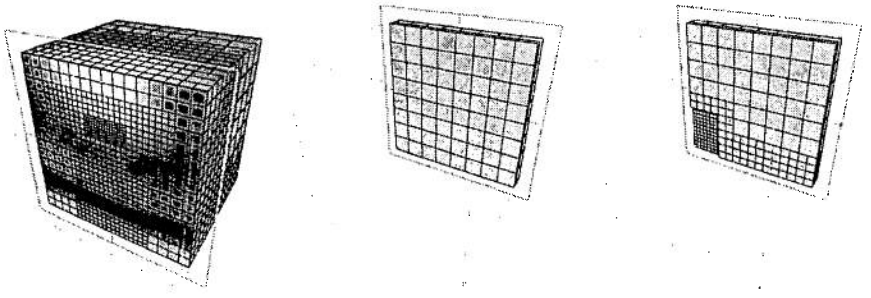


FIG. 4.6. *Problem (4.4). Slices through the mesh perpendicular to the x-axis. Left: Correlation strategy. Slice near* x_2 *at* $t = 10$ *(block 8). Middle: Correlation strategy. Slice near* x_1 *at* $t = 8.8$ *(block 7). Right: Memory bound strategy. Slice near* x_1 *at* $t = 8.8$ *(block 4).*

Block	T_b	T_{b+1}	# vertices	# hexahedra
1	0	0.09	58799	51066
2	0.09	0.15	58424	50289
3	0.15	0.27	58393	50359
4	0.27	0.61	59102	50744
5	0.61	0.99	28395	23388

TABLE 4.4
Problem (4.5). Blocks resulting from the memory bound strategy.

We provide "slices" through the mesh that are perpendicular to the x-axis at $x = 0.5$ for four representative times in Fig. 4.7.

5. Conclusions. In this paper, we consider adaptive algorithms for evolution problems that use a sequence of "blocks" in time which employ fixed, non-uniform space meshes. Blockwise adaptive algorithms provide a way to balance the goal of achieving desired accuracy using discretizations with relatively few degrees of freedom with the computational overhead associated with load balancing, re-meshing, matrix reassembly and error estimation. Block adaptive algorithms achieve this balance by minimizing the number of mesh changes. However, a major issue is determining block discretizations from coarse scale solution information that achieve the desired accuracy and efficiency. We describe several strategies to achieve this goal using adjoint-based *a posteriori* error estimates. We demonstrate the behavior of the proposed algorithms as well as several technical issues in a set of examples.

REFERENCES

[1] I. BABUŠKA AND T. STROUBOULIS, *The Finite Element Method and its Reliability*, Clarendon Press, New York, 2001.
[2] W. BANGERTH AND R. RANNACHER, *Adaptive Finite Element Methods for Differential Equations*, Birkhauser Verlag, 2003.
[3] R. BECKER AND R. RANNACHER, *An optimal control approach to* a posteriori *error estimation in finite element methods*, Acta Numerica, (2001), pp. 1–102.
[4] M. BERGER AND P. COLELLA, *Local adaptive mesh refinement for shock hydrodynamics*, J. Comput. Phys., 82 (1989), pp. 64–84.
[5] M. BIETERMAN, J. FLAHERTY, AND P. MOORE, *Adaptive refinement methods for non-linear parabolic partial differential equations*, in Accuracy Estimates and Adaptive Refinements in Finite Element Computations, John Wiley and Sons, Ltd., New York, 1986.
[6] V. CAREY, D. ESTEP, AND S. TAVENER, *Averaging based projections in operator decomposition methods for elliptic systems.* In preparation, 2009.
[7] S. F. DAVIS AND J. E. FLAHERTY, *An adaptive finite element method for initial-boundary value problems for partial differential equations*, SIAM J. Sci. Stat. Comput., 3 (1982), pp. 85–107.
[8] T. DUPONT, *Mesh modification for evolution equations*, Math. Comp., 39 (1982), pp. 85–107.
[9] K. ERIKSSON, D. ESTEP, P. HANSBO, AND C. JOHNSON, *Introduction to adaptive methods for differential equations*, Acta Numerica, (1995), pp. 105–158.
[10] ———, *Computational Differential Equations*, Cambridge University Press, New York, 1996.
[11] K. ERIKSSON AND C. JOHNSON, *Adaptive finite element methods for parabolic problems I: A linear model problem*, SIAM J. Numer. Anal., 28 (1991), pp. 43–77.
[12] D. ESTEP, *A posteriori error bounds and global error control for approximations of ordinary differential equations*, SIAM J. Numer. Anal., 32 (1995), pp. 1–48.
[13] D. ESTEP AND D. FRENCH, *Global error control for the continuous Galerkin finite element method for ordinary differential equations*, RAIRO Modél. Math. Anal. Numér., 28 (1994), pp. 815–852.

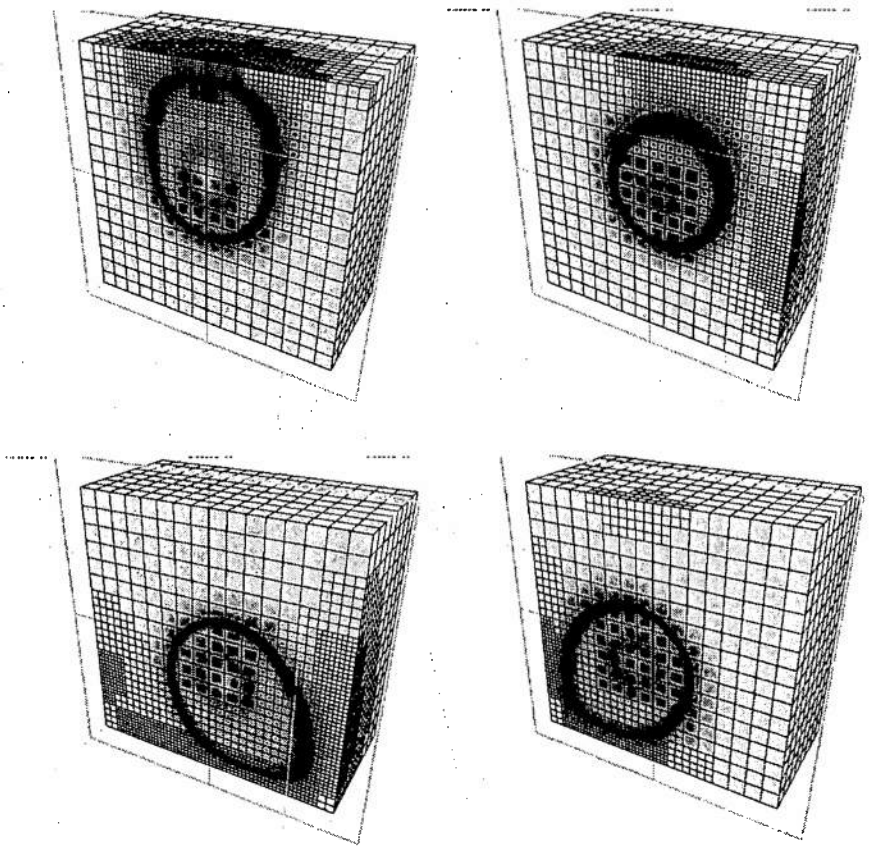


FIG. 4.7. *Problem (4.5). Memory bound strategy. Slices through the mesh perpendicular to the x-axis at* $x = 0.5$. *Upper left:* $t = 0.04$ *(block 1). Upper right:* $t = 0.16$ *(block 3). Lower left:* $t = 0.42$ *(block 4). Lower right:* $t = 0.62$ *(block 5).*

[14] D. ESTEP, M. HOLST, AND M. LARSON, *Generalized Green's functions and the effective domain of influence*, SIAM J. Sci. Comput., 26 (2005), pp. 1314–1339.

[15] D. ESTEP, M. LARSON, AND R. WILLIAMS, *Estimating the error of numerical solutions of systems of reaction-diffusion equations*, Memoirs A.M.S., 146 (2000), pp. 1–109.

[16] D. ESTEP AND J. SANDELIN, *Cancelation of error and adaptive error control for ordinary differential equations*. In preparation, 2009.

[17] D. ESTEP AND R. WILLIAMS, *Accurate parallel integration of large sparse systems of differential equations*, Math. Models Meth. Appl. Sci., 6 (1996), pp. 535–568.

[18] R. E. EWING, R. D. LAZAROV, AND A. T. VASSILEV, *Adaptive techniques for time-dependent problems*, Comput. Methods Appl. Mech. Engrg., 101 (1992), pp. 113–126.

[19] J. E. FLAHERTY, R. M. LOY, C. OZTURAN, M. S. SHEPHARD, B. K. SZYMANSKI, J. D. TERESCO, AND L. H. ZIANTZ, *Parallel structures and dynamic load balancing for adaptive finite element computation*, Appl. Numer. Math., 26 (1998), pp. 241–263.

[20] M. GILES AND E. SÜLI, *Adjoint methods for PDEs: A posteriori error analysis and postprocessing by duality*, Acta Numerica, (2002), pp. 145–236.

[21] E. HAIRER, S. NORSETT, AND G. WANNER, *Solving Ordinary Differential Equations I*, Springer-Verlag, New York, 1987.

[22] P. K. MOORE AND J. E. FLAHERTY, *Adaptive local overlapping grid methods for parabolic-systems in 2 space dimensions*, J. Comput. Phys., 98 (1992), pp. 54–63.
[23] J.-F. REMACLE, O. KLAAS, J. E. FLAHERTY, AND M. S. SHEPHARD, *Parallel algorithm oriented mesh database*, Engineering with Computers, 18 (2002), pp. 274–284.
[24] J. R. STEWART AND H. C. EDWARDS, *Parallel adaptive application development using the SIERRA framework*, in Proceedings of the First MIT Conference in Computational Fluid and Solid Mechanics, Elsevier, Amsterdam, 2001.
[25] T. WILDEY, *Adaptive Coupled Equation Solver (ACES)* `http://users.ices.utexas.edu/~twildey/software`.

Appendix A. Detailed description of a block adaptive algorithm.

The notation used in our block adaptive algorithm is as follows.

1. `Ntimestep` = current number of time steps
2. `Nelem(j)` = number of space elements in timestep j, i.e., for $t \in [t_{j-1}, t_j]$
3. `Ntimestep_children(j)` = number of subintervals into which timestep j is to be divided
4. `Nelem_children(i,j)` = number of subelements into which finite element i is to be divided in timestep j
5. The bth "block" is time interval $[T_{b-1}, T_b] = [t_{b,0}, t_{b,N_b}]$
6. The bth "block" comprises timesteps $j_{b-1}, \ldots, j_b$, i.e., $N_b = j_b - j_{b-1}$, $t_{b,0} = t_{j_{b-1}}$ and $t_{b,N_b} = t_{j_b}$.
7. `block(i,b)` = number of intervals the parent element i will be divided into on block b.
8. `Nelem_block(b)` = number of elements in block b.
9. We use the MATLAB colon operator : to denote the full row or column.
10. The parameter θ governs how often a mesh is coarsened; $\theta \approx 10$ works well.

Algorithm A.1 A memory-bound strategy

Input error tolerance `TOL`, maximum number of elements in any block `Nmax`, the initial coarse-scale discretization for the forward problem, and the coarse-scale discretization for the adjoint problems
Solve forward problem (1.1) for U on $[0,T]$
Project forward solution onto coarse-scale adjoint problem mesh
Solve adjoint problem (2.4) on coarse scale mesh and compute $E(U)$
Compute LATOL, $Æ_x$, $Æ_t$ (3.6),(3.5)
for $j = 1,\ldots,$`Ntimesteps` **do**
    Compute `Ntimestep_children(j)` (3.13)
    for $i = 1,\ldots,$`Nelem(j)` **do**
        Compute `Nelem_children(i,j)` (3.16)
    end for
end for
`Ntimesteps` $\leftarrow \sum_{j=1}^{\texttt{Ntimesteps}}$ `Ntimestep_children(j)`
Each subinterval of $[t_{j-1}, t_j]$ inherits `Nelem_children(i,j)`
$b = 1$, $T_0 = 0$, $T_1 = k_1$, $j_0 = 1$, $j = 2$
`block(:,b)` $\leftarrow$ `Nelem_children(:,1)`
`Nelem_block(b)` $\leftarrow \sum_i$ `block(i,b)`
while $T_b < T$ **do**
    while `Nelem_block(b)` < `Nmax` and
            `Nelem_block(b)` $< \theta \times \sum_{i=1}^{\texttt{Nelem(j)}}$ `Nelem_children(i,j)` **do**
        $j_b \leftarrow j$
        $T_b \leftarrow T_b + k_j$
        `block(:,b)` $\leftarrow$ max[`block(:,b)`, `Nelem_children(:,j)`]
        `Nelem_block(b)` $= \sum_i$ `block(i,b)`
        $j \leftarrow j + 1$
    end while
    $b \leftarrow b + 1$
end while
for $i = 1,\ldots,b$ **do**
    Compute new mesh for block b
    Optional: Estimate projection error and correct predicted meshes if necessary
end for
for $i = 1,\ldots,b$ **do**
    Solve forward problem on block b for U
    Project U onto mesh for block $b+1$
    Optional: Compute projection error between blocks and correct meshes
end for

To implement the correlation-based strategy, we alter the block selection criteria ($\sum$ `block`$(b) \leq$ `Nmax`) with a step which accepts a block if `block`$(:,b)$ is correlated to `Nelem_children`$(:,j)$ and `Nelem_block(b)` is less than `Nmax`.

The algorithm assumes that the blocks are always generated (even on repeat solve cycles) using the coarse mesh as a base. The algorithm may be easily modified to work recursively on the blocks. It may also be modified, with a little more care, to allow merging and splitting of blocks during repeated solves.

1

ERROR ESTIMATES FOR MULTISCALE OPERATOR DECOMPOSITION FOR MULTIPHYSICS MODELS

1.1 Introduction

Multiphysics, multiscale models that couple different physical processes acting across a large range of scales are encountered in virtually all scientific and engineering applications. Such systems present significant challenges in terms of computing accurate solutions and for estimating the error in information computed from numerical solutions. In this chapter, we discuss the problem of computing accurate error estimates for one of the most common, and powerful, numerical approaches for multiphysics, multiscale problems.

1.1.1 *Examples of multiphysics models*

Without any attempt to be complete, we describe three examples of multiphysics models that illustrate some different ways in which physical processes may be coupled.

Example 1.1 A thermal actuator A thermal actuator is a MEMS (microelectronic mechanical switch) device. A contact rests on thin braces composed of a conducting material. When a current is passed through the braces, they heat up and consequently expand to close the contact, see Fig. 1.1. The actuator is modeled by a system of three coupled equations, each representing a distinct

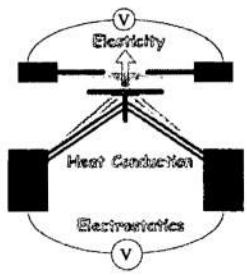


FIG. 1.1. Sketch of a thermal actuator.

physical process acting on its own scale. The first is an electrostatic current equation

$$\nabla \cdot (\sigma \nabla u_1) = 0, \tag{1.1}$$

governing potential u_1 (where the current is $J = -\sigma \nabla u_1$), the second is a steady-state energy equation

$$\nabla \cdot (\kappa(u_2)\nabla u_2) = \sigma(\nabla u_1 \cdot \nabla u_1), \tag{1.2}$$

for the governing temperature u_2, and a linear elasticity equation giving the steady-state displacement u_3,

$$\nabla \cdot \left(\lambda \mathrm{tr}(E) I + 2\mu E - \beta(u_2 - u_{2,ref})I\right) = 0, \quad E = \left(\nabla u_3 + \nabla u_3^{\top}\right)/2. \tag{1.3}$$

This is an example of "parameter passing", in which the solution of one component is used to compute the parameters and/or data for another component. Note that the electric potential u_1 can be calculated independently of u_2 and u_3. The temperature u_2 can be calculated once the electric potential u_1 is known, while the calculation of displacement u_3 requires prior knowledge of u_2, and therefore of u_1.

Example 1.2 The Brusselator problem First introduced by Prigogine and Lefever (Prigogine and Lefever, 1968) as a model of chemical dynamics, the Brusselator problem consists of a coupled set of equations,

$$\begin{cases} \frac{\partial u_1}{\partial t} - k_1 \frac{\partial^2 u_1}{\partial x^2} = \alpha - (\beta + 1)u_1 + u_1^2 u_2, & x \in (0,1), t > 0, \\ \frac{\partial u_2}{\partial t} - k_2 \frac{\partial^2 u_2}{\partial x^2} = \beta u_1 - u_1^2 u_2, & x \in (0,1), t > 0, \\ u_1(0,t) = u_1(1,t) = \alpha, \; u_2(0,t) = u_2(1,t) = \beta/\alpha, & t > 0, \\ u_1(x,0) = u_{1,0}(x), \; u_2(x,0) = u_{2,0}(x), & x \in (0,1), \end{cases} \tag{1.4}$$

where u_1 and u_2 are the concentrations of species 1 and 2, respectively. Solutions of the Brusselator problem exhibit a wide range of behavior depending on parameter values.

Reaction-diffusion equations are an example of a problem that combines different physics - in this case, reaction and diffusion - in one equation. The generic picture for a reaction-diffusion equation is a relatively fast, destabilizing reaction component interacting with a relatively slow, stabilizing diffusion component. Thus, the physical components have both different scales and different stability properties.

Example 1.3 Conjugate heat transfer between a fluid and solid object We consider the flow of a heat-conducting Newtonian fluid past a solid cylinder as shown in Fig. 1.2.

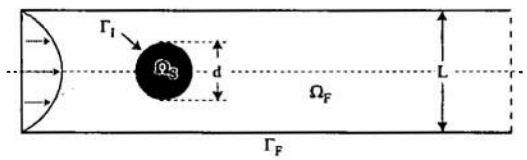


FIG. 1.2. Computational domain for flow past a cylinder

The model consists of the heat equation in the solid and the equations governing the conservation of momentum, mass and energy in the fluid, where we apply the Boussinesq approximation to the momentum equations in the fluid. The temperature field is advected by the fluid and couples back to the momentum equations through the buoyancy term.

Let Ω_F and Ω_S be polygonal domains in $\mathbb{R}^2$ with boundaries $\partial\Omega_F$ and $\partial\Omega_S$ intersecting along an interface $\Gamma_I = \partial\Omega_S \cap \partial\Omega_F$. The complete coupled problem is

$$\begin{cases} -\mu\Delta u + \rho_0 (u \cdot \nabla) u + \nabla p + \rho_0 \beta T_F g = \rho_0 (1 + \beta T_0) g, & x \in \Omega_F, \\ -\nabla \cdot u = 0, & x \in \Omega_F, \\ -k_F \Delta T_F + \rho_0 c_p (u \cdot \nabla T_F) = Q_F, & x \in \Omega_F, \\ \begin{cases} T_S = T_F, \\ k_F (n \cdot \nabla T_F) = k_S (n \cdot \nabla T_S), \end{cases} & x \in \Gamma_I, \\ -k_S \Delta T_S = Q_S, & x \in \Omega_S, \end{cases} \tag{1.5}$$

where ρ_0 and T_0 are reference values for the density and temperature respectively, μ is the molecular viscosity, β is the coefficient of thermal expansion, c_p is the specific heat, k_F and k_S are the thermal conductivities of the fluid and solid respectively, Q_F and Q_S are source terms and n is the unit normal vector directed into the fluid. Note that u is a vector.

We define $\Gamma_{u,D}$ and $\Gamma_{u,N}$ to be the boundaries on which we apply Dirichlet and Neumann conditions for the velocity field respectively, and set

$$\begin{cases} u = g_{u,D}, & x \in \Gamma_{u,D}, \\ \mu \dfrac{\partial u}{\partial n} = g_{u,N}, & x \in \Gamma_{u,N}. \end{cases}$$

Similarly, we define $\Gamma_{T_F,D}$, $\Gamma_{T_F,N}$, $\Gamma_{T_S,D}$, and $\Gamma_{T_S,N}$ to be the boundaries on which we impose Dirichlet and Neumann conditions for the temperature fields in the fluid and the solid respectively, and set

$$\begin{cases} T_F = g_{T_F,D}, & x \in \Gamma_{T_F,D}, \\ k_F (n \cdot \nabla T_F) = g_{T_F,N}, & x \in \Gamma_{T_F,N}, \\ T_S = g_{T_S,D}, & x \in \Gamma_{T_S,D}, \\ k_S (n \cdot \nabla T_S) = g_{T_S,N}, & x \in \Gamma_{T_S,N}. \end{cases}$$

This presents a class of problems where different physics in different physical domains are coupled through interactions across a common boundary.

1.1.2 *Challenges and goals of multiscale, multiphysics models*

Multiscale, multiphysics models are characterized by intimate interactions between different physics across a wide range of scales. This poses challenges for solving such problems, e.g.

Accurate and efficient computation Computing information that depends on solution behavior occurring at very different scales is problematic. It is rarely possible to simply to use a discretization sufficiently fine to resolve the finest scale behavior.

Complex stability A multiphysics model generally offers a complex stability picture that results from a fusion of the stability properties of different physics. For example, consider a reacting fluid that combines fluid flow with the dynamical properties of reaction-diffusion equations.

Linking different physics across scales Understanding the significance of linkages between physical components and how those affect model output is another complicated issue. In many situations, the output of one physical component must be transformed and/or scaled to obtain information relevant to the other components.

Another complication is the range of applications of multiphysics models. These include

Model prediction Perhaps the chief goal of mathematical modeling is to predict the behavior of the modeled system outside of the range of physical observation.

Sensitivity analysis The reliability of model predictions depends on analyzing the effects of uncertainties and variation in the physical properties of the model on its output.

Parameter optimization In design problems, the goal is to determine optimal parameter values with respect to producing a desired observation or consequence.

Such applications require computation of solutions corresponding to wide range of data and parameters. We expect the solution behavior to vary significantly and the ability to obtain accurate numerical solutions therefore to vary as well. This raises a critical need for quantification and control of numerical error.

The solution and application of multiphysics, multiscale models invoke two computational goals:

- Compute specific information from multiscale, multiphysics models accurately and efficiently
- Accurately quantify the error and uncertainty in any computed information

The context is important:

It is often difficult or impossible to obtain solutions of multiscale, multiphysics models that are uniformly accurate throughout space and/or time

Thus, we are interested in computing accurate error estimates for solutions that are relatively inaccurate. This is an important consideration, given that much of classical error analysis is derived under conditions that amount to assuming that the numerical solution is in the "asymptotic range of convergence", meaning that the solution is sufficiently accurate that the rate of convergence can be observed

by uniform refinement of the discretization. It is rarely possible to reach this level of discretization in a multiphysics, multiscale problem.

1.1.3 *Multiscale, multidiscretization operator decomposition*

Multiscale, multidiscretization operator decomposition is a widely used technique for solving multiphysics, multiscale models. The general approach is to decompose the multiphysics and/or multiscale problem into components involving simpler physics over a relatively limited range of scales, and then to seek the solution of the entire system through some sort of iterative procedure involving numerical solutions of the individual components. We illustrate in Fig. 1.3. In general, different components are solved with different numerical methods as well as with different scale discretizations. This approach is appealing because there is generally a good understanding of how to solve a broad spectrum of single physics problems accurately and efficiently, and because it provides an alternative to accommodating multiple scales in one discretization.

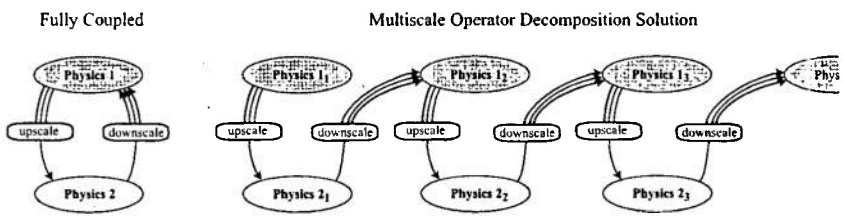


FIG. 1.3. Left: Illustration of a multiscale, multiphysics model. Right: Illustration of a multiscale operator decomposition solution.

Example 1.4 A classic example of multiscale operator decomposition is operator splitting for a reaction-diffusion equation,

$$\begin{cases} \dfrac{\partial u}{\partial t} = \epsilon \Delta u + f(u), & x \in \Omega, 0 < t, \\ \text{suitable boundary conditions}, & x \in \partial\Omega, 0 < t, \\ u(\cdot, 0) = u_0(\cdot) \end{cases} \tag{1.6}$$

where $\Omega \subset \mathbb{R}^d$ is a spatial domain and f is a smooth function. Accuracy considerations dictate the use of relatively small steps to integrate a fast reaction component. On the other hand, stability considerations over moderate to long time intervals suggests the use of implicit, dissipative numerical methods for integrating diffusion problems. Such methods are expensive to use per step, but relatively large steps can be used on a purely dissipative problem. If the reaction and diffusion components are integrated together, then the small steps required for accurate resolution of the reaction lead to an expensive computation.

In a multiscale operator splitting approach, the reaction and diffusion components are integrated independently inside each time interval of a discretization

of time and "synchronized" in some fashion only at the nodes of the interval. The reaction component is often integrated by using significantly smaller sub-steps (e.g. 10^{-5} smaller is not uncommon) than those used to integrate the diffusion component, which can lead to a tremendous computational savings.

Employing the method of lines, if we discretize in space using a continuous, piecewise linear finite element method with M elements, see Sec. 1.4.1, we obtain the initial value problem: find $y \in \mathbb{R}^M$ such that

$$\begin{cases} \dot{y} = Ay(t) + F(y(t)), & 0 < t \leq T, \\ y(0) = y_0, \end{cases} \tag{1.7}$$

where A is an $l \times l$ constant matrix representing a "diffusion component" and $F(y) = (F_1(y), F_2(y), \cdots, F_l(y))^{\mathsf{T}}$ is a vector of nonlinear functions representing a "reaction component".

We first discretize $[0, T]$ into $0 = t_0 < t_1 < t_2 < \cdots < t_N = T$ with diffusion time steps $\{\Delta t_n\}_{n=1}^N$, $\Delta t_n = t_n - t_{n-1}$, and $\Delta t = \max_{1 \leq n \leq N}(\Delta t_n)$. We define a piecewise continuous approximate solution

$$\tilde{y}(t) = \frac{t_n - t}{\Delta t_n}\, \tilde{y}_{n-1} + \frac{t - t_{n-1}}{\Delta t_n}\, \tilde{y}_n, \quad t_{n-1} \leq t \leq t_n, \tag{1.8}$$

with nodal values $\{\tilde{y}_n\}$ obtained from the following procedure:

Algorithm 1 Abstract Operator Splitting for Reaction-Diffusion Equations

Set $\tilde{y}_0 = y_0$
for $n = 1, \cdots, N$ **do**
Compute $y^r(t_n^-)$ satisfying the reaction component

$$\begin{cases} \dot{y^r} = f(y^r(t)), & t_{n-1} < t \leq t_n, \\ y^r(t_{n-1}^+) = \tilde{y}_{n-1} \end{cases} \tag{1.9}$$

Compute $y^d(t_n^-)$ satisfying the diffusion component

$$\begin{cases} \dot{y^d} = Ay^d(t), & t_{n-1} < t \leq t_n, \\ y^d(t_{n-1}^+) = y^r(t_n^-) \end{cases} \tag{1.10}$$

Set $\tilde{y}_n = y^d(t_n^-)$
end for

With a little thought, we recognize that this algorithm has the potential to be a multiscale solution procedure since we can now resolve the solution of each component on independent scales. That is one benefit of using operator decomposition. Unfortunately, this decomposition has unforseen effects on both accuracy and stability. The reason is that we have discretized the instantaneous interaction between the reaction and diffusion components.

Example 1.5 In (Estep *et al.*, 2008*a*), we consider a problem in which the reaction component exhibits finite time blow up when undamped by the diffusion component. The problem is

$$\begin{cases} \dot{y} + \lambda y = y^2, & t > 0, \\ y(0) = y_0 \in \mathbb{R}, \end{cases} \tag{1.11}$$

which has exact solution

$$y(t) = \frac{\lambda y_0}{y_0 - (y_0 - \lambda)\, e^{\lambda t}}, \tag{1.12}$$

when $\lambda \neq 0$. The exact solution exists for all time and tends to zero as $t \to \infty$ when $\lambda > y_0$. On the other hand, there is finite time blow up, e.g. $y \to \infty$ at a finite time, if $\lambda < y_0$.

Applying the operator splitting to (1.11), the solutions of the two components and the operator splitting solution are,

$$y^r(t) = \frac{y^{d-}_{n-1}}{1 - y^{d-}_{n-1}(t - t_{n-1})}, \quad y^d(t) = e^{-\lambda(t-t_{n-1})}\, y^{r-}_n, \quad \tilde{y}_n = \frac{e^{-\lambda \Delta t_n} \tilde{y}_{n-1}}{1 - \Delta t_n\, \tilde{y}_{n-1}},$$

when the reaction component is defined. We see that decoupling the smoothing effect provided by instantaneous interaction with the diffusion component means that the reaction component can blow up in finite time. This has an effect on numerical solution.

We consider the time steps introduced above, $\{\Delta t_n\}_{n=1}^N$, to be diffusion time steps. For each diffusion step, we choose a (small) time step $\Delta s_n = \Delta t_n / M_n$ with $\Delta s = \max_{1 \le n \le N} \Delta s_n$, and the nodes $t_{n-1} = s_{0,n} < s_{1,n} < \cdots < s_{M_n,n} = t_n$ (see Fig. 1.4). We associate the time intervals $I_n = [t_{n-1}, t_n]$ and $I_{m,n} = [s_{m-1,n}, s_{m,n}]$ with these discretizations. Without going into details, we solve

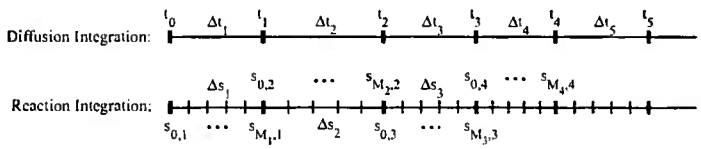


FIG. 1.4. Discretization of time used for multiscale operator splitting

the components (1.9), (1.10) using the forward and backward Euler method respectively,

$$Y^{r-}_{m,n} = Y^{r-}_{m-1,n} + f(Y^{r-}_{m-1,n})\, \Delta s_n, \quad Y^{d-}_n = Y^{d-}_{n-1} + A Y^{d-}_n\, \Delta t_n.$$

See Sec. 1.4.4 for details on discretization of evolution problems. We compute a piecewise linear discrete approximation $\tilde{Y}$ using the nodal values of Y^d.

On the left side of Fig. 1.5, we plot the true solution and the nodal values of the approximation $\tilde{Y}$ computed with $N = 50$ diffusion steps and $M = 1$ reaction step per diffusion step. The approximation is reasonably accurate. Next,

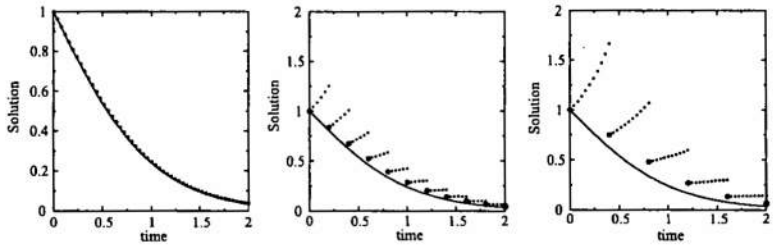


FIG. 1.5. Plots of the approximation $\tilde{Y}$ and the true solution. Left: $N = 50$, $M = 1$. Middle: $N = 10$, $M = 5$. Right: $N = 5$, $M = 10$. The nodal values of $\tilde{Y}$ are denoted by the larger points while the smaller points denote node values of the reaction component Y^r.

we show the results when the diffusion step is increased by choosing $N = 10$ and, in order to maintain the same resolution of the reaction component, we correspondingly increase to $M = 5$ reaction steps per diffusion step. The node values of $\tilde{y}$ are relatively close to those of y. The subsequent nodal values of the reaction component solution y^r inside each step move away from the true solution. This large departure is somewhat counteracted by application of the diffusion operator. The reaction components exhibit significant growth inside each diffusion step, which severely affects accuracy.

If we increase the diffusion step by taking $N = 5$ and maintain resolution in the reaction component by taking $M = 10$, the approximation becomes even less accurate. If we increase the diffusion step further, then the reaction component actually blows up inside a diffusion step.

We emphasize that the error in this example is a consequence of a kind of instability introduced by multiscale operator decomposition. We will see below that multiscale operator decomposition commonly affects both accuracy and stability in a wide variety of problems.

1.2 The key is stability. But what is stability ... and stability of what?

Stability is likely one of the most shopworn terms in mathematics; given so many meanings so as to cause a high probability of mis-communication in any mixed crowd. Nonetheless, stability is the key to quantifying the effects of error and uncertainty on the output of a computed model solution.

Generally, it is impossible to give a definitive definition of stability in the context of a multiphysics model. A computational mathematician might think

in terms of numerical stability, with instability characterized by oscillations on the scale of the discretization. A mathematician in partial differential equations might be thinking in terms of well-posedness, i.e. continuous dependence on data. A physicist might be worried about preserving the conservation of important quantities like mass and energy. A dynamicist might think in terms of the stability properties of stationary solutions and attracting manifolds.

Indeed, all of these views of stability, and likely others, are important in the right contexts. In fact, the only definitive thing to be said about stability is that it is very unlikely that just one view of stability will suffice when solving a multiphysics, multiscale problem.

1.2.1 *Pointwise stability of the Lorenz problem*

To illustrate the complexity of stability, we consider the infamous Lorenz problem,

$$\begin{cases} \dot{u}_1 = -10u_1 + 10u_2, \\ \dot{u}_2 = 28u_1 - u_2 - u_1u_3, \quad 0 < t, \\ \dot{u}_3 = -\frac{8}{3}u_3 + u_1u_2. \end{cases} \tag{1.13}$$

The Lorenz equations were derived by Lorenz (Lorenz, 1963) as a gross simplification of a weather model to explain why weather predictions become inaccurate after a few days. We have chosen parameter values believed to lead to chaotic behavior. In Fig. 1.6 we plot a solution. All solutions rapidly approach the "strange

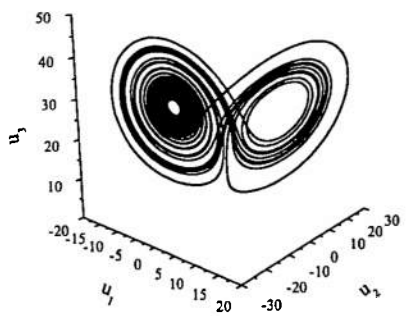


FIG. 1.6. Solution of the Lorenz problem (1.13) corresponding to initial condition $(-9.408, -9.096, 28.581)$.

attractor" where they subsequently remain. The dynamical behavior is always the same in qualitative terms. There are two non-zero steady state solutions and a generic solution is either "orbiting" one of these solutions or transitioning between orbits. The orbits spiral away from the steady-state solution at the center until a point when a solution is sufficiently far away from the fixed point, whereupon it moves to orbit around the other fixed point. In a crude way, solutions behave in a very predictable fashion.

Chaos is often described as "sensitivity to initial conditions", which means that solutions that begin close by to each other eventually move apart. In Fig. 1.7, we plot a second solution to the Lorenz problem that begins near the solution plotted in Fig. 1.6 along with the distance between the solutions. The two solu-

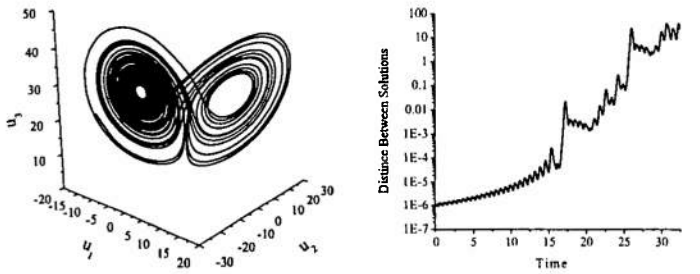


FIG. 1.7. A second solution of the Lorenz problem (1.13) corresponding to initial condition $(-9.408, -9.096001, 28.581)$ and the pointwise difference between the two solutions.

tions start close together and actually remain close until around time 17.5, when there is a rapid increase in their separation. For a brief period between 18 and 21, the separation remains fairly constant, and then it begins to increase again, with another very rapid increase around 24. All solutions must remain in a compact region around the origin, so at some point the distance between the solutions reaches the order of size of the compact region and cannot grow further.

We conclude that two solutions that are pointwise close at some time eventually diverge pointwise at a later time. The chaotic nature of the Lorenz problem means that it is difficult to predict the pattern of orbits around the fixed points with any accuracy very far into the future. On the other hand, nearby solutions may remain close for quite some time and may even become closer before eventually diverging.

The source of chaotic behavior in the Lorenz problem is actually rather complex, (Estep and Johnson, 1998). However, one important factor is relatively easy to explain. Following (Estep and Johnson, 1998), in Fig. 1.8, we show a plot looking straight down the vertical axis at parts of many solutions. The solutions shown in the lower left corner are orbiting around one of the nonzero fixed points or, if they are in the "outer" orbit, moving to the neighborhood of the other fixed point. Likewise, the solutions plotted in the upper right corner are either orbiting around a fixed point or moving to a neighborhood of the other fixed point.

We note that there are two solutions in the lower left-hand region, that are very close until they approach the vertical u_3 axis but then rapidly move apart after that. In fact, there is a separatix, or manifold, coming out of the u_3 axis that separates solutions taking another orbit around a fixed point from those

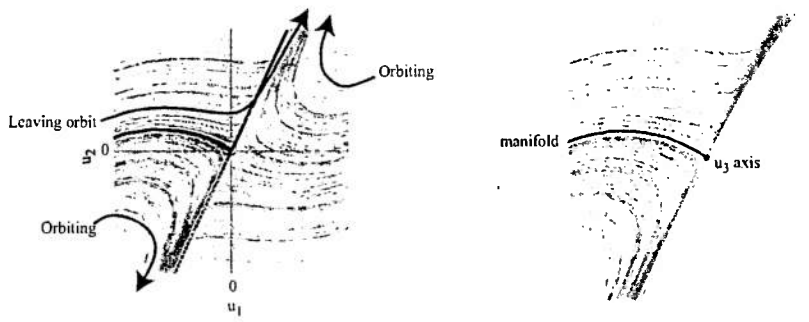


FIG. 1.8. Left: Looking straight down the u_3 (vertical) axis at many solutions of the Lorenz equations (1.13). Solutions passing through a neighborhood of a separatix coming out of the u_3 axis, shaded in the figure, are very sensitive to small perturbations while they are in that neighborhood. Right: A plot of the separatix.

that transition to the other fixed point. Solutions on either side of this manifold move apart rapidly. Thus, we see how small perturbations can lead to rapid separation. Any solution that passes near the neighborhood of the separatix, shaded in the figure, become very sensitive to small perturbations during the short time the solution remains in the neighborhood. Eventually, all solutions pass nearby the separatix and thus become sensitive to perturbation. Away from the neighborhood of the separatix, the distance between solutions may grow or shrink slowly, e.g. at a polynomial rate. This explains the pattern of separation seen in the plot of distance between two solutions in Fig. 1.7.

Not surprisingly, chaotic behavior affects numerical solutions as well. In Fig. 1.9, we show the effects of varying step sizes on pointwise accuracy. We

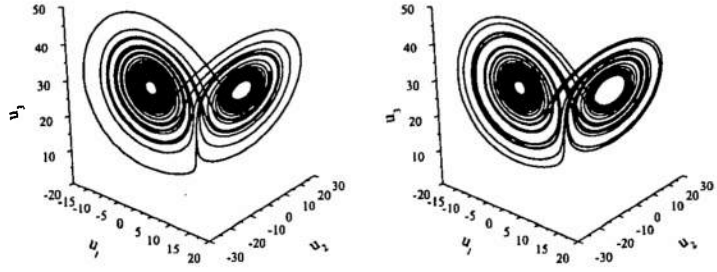


FIG. 1.9. Two numerical approximations of the Lorenz solution shown in Fig. 1.6 are shown; the solution on the left is accurate while the solution on the right is computed with larger step sizes. The distance between the solutions is shown on the right.

plot the difference between the numerical solutions on the left in Fig. 1.10. The

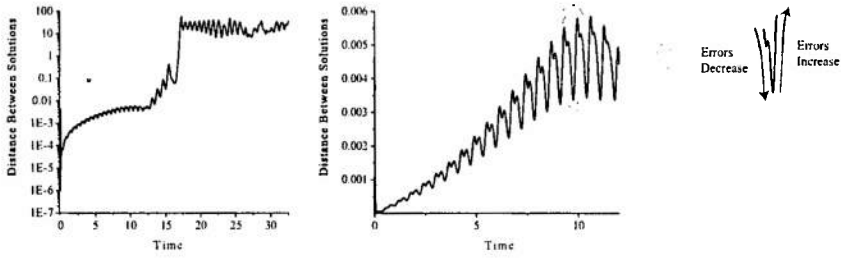


FIG. 1.10. Left: Plot of the pointwise difference between the numerical solutions of Lorenz shown in Fig. 1.9. Right: A blowup of the difference for $0 \leq t \leq 12$.

pointwise numerical error clearly follow an increasing trend, but it does not increase monotonically. In fact, the pointwise error actually decreases during some short periods of time, see the plot on the right in Fig. 1.10.

1.2.2 *Classic a priori stability analysis*

But what about the classic *a priori* stability analysis that is taught in courses in differential equations and numerical analysis (*a priori* means that it is carried out before any solutions are computed)? Do these classic notions of stability present a reasonable picture for a particular solution? In fact, the answer is decidedly no in most cases. The classic view of stability for linear problems tends to be black and white; a particular solution is either stable or unstable with respect to perturbations. We see that this point of view fails for solutions of simple nonlinear problems, e.g. the Lorenz problem.

Example 1.6 It is easy to illustrate the shortcomings of *a priori* stability analysis using error analysis of the numerical solution of a matrix system of equations. Consider a numerical solution $Y \approx y$ of a matrix system

$$\mathbf{A}y = b, \tag{1.14}$$

computed using Gaussian elimination. The computable residual of Y is $R = \mathbf{A}Y - b$ and the classic relative error bound ((Higham, 2002)) is

$$\frac{\|Y - y\|}{\|y\|} \leq C\kappa(\mathbf{A})\frac{\|R\|}{\|b\|}, \tag{1.15}$$

where the condition number $\kappa(\mathbf{A}) = \|\mathbf{A}\|\,\|\mathbf{A}^{-1}\|$ is a measure of the sensitivity of the solution of (1.14) to perturbations in the data, i.e. the stability properties of the inverse operator of $\mathbf{A}$. In particular,

$$\kappa(\mathbf{A}) = \frac{\|\mathbf{A}\|}{distance\ from\ \mathbf{A}\ to\ \{singular\ matrices\}}.$$

(To be precise, we have to specify norms, but that level of detail is not important here.)

We now solve (1.14) using Gaussian elimination, where $\mathbf{A}$ is a random 800×800 matrix, where the coefficients in the random matrix are uniformly distributed $U(-1,1)$ on $(-1,1)$. The goal is to determine the first component y_1 of the solution. The condition number of $\mathbf{A}$ is 6.7×10^4. Straightforward computation yields

$$\text{actual error in the quantity of interest} \approx 1.0 \times 10^{-15},$$
$$\text{traditional error bound for the error} \approx 3.5 \times 10^{-5}$$

We see that the traditional error bound is orders of magnitude larger than the actual error and is essentially useless as far as estimating the error in the particular computed information.

We remark that the bound (1.15) is a specific example of a general "meta-theorem", which reads

Theorem 1.7

$$\|\textit{effect of perturbation on the output of an operator}\|$$
$$\leq \|\textit{measure of stability of the operator}\| \times \|\textit{size of the perturbation}\|. \quad (1.16)$$

In the linear algebra example, we have

$$AY = b + R,$$

hence we can think of the numerical solution Y as solving the linear system (1.14) with perturbed data $b + R$.

The pessimism of a classic *a priori* stability bound is not surprising given a little reflection. After all, the goal of such a bound is to account for the largest possible error in a large class of solutions corresponding to a large set of data, not produce an accurate error estimate for particular information computed from a particular solution. The power of an *a priori* error bound is that it characterizes the general behavior of the numerical method.

The situation for nonlinear problems is worse. For example, in nonlinear evolution problems, the classic stability analysis uses a Gronwall argument to obtain a bound in the form,

$$\text{effect of perturbation at time } t \leq C\, e^{Lt} \times \text{size of perturbation},$$

where C and L are constants with L typically large (L is on the order of 100 in the Lorenz example). Such bounds are non-descriptive past a very short initial transient, e.g. even for the chaotic Lorenz problem. The factor Ce^{Lt} plays the role of a condition number (for an absolute error estimate).

1.2.3 *Stability for stationary problems*

There is a long tradition of conducting careful, precise analysis of stability for evolutionary problems and, in particular, distinguishing different types of stability properties. It is perhaps fair to say that stability for stationary problems tends to be treated more crudely, at least for elliptic problems. This is unfortunate because stability is just as complex an issue for stationary problems as evolutionary problems.

Example 1.8 To illustrate this claim, we consider an elliptic problem

$$\begin{cases} L(u) = -\nabla \cdot \left((.001 + |\tanh(10(y+1)|)\nabla u \right) \\ \qquad\qquad -\alpha \times \begin{pmatrix} 50(x-1.5) \\ 50 \end{pmatrix} \cdot \nabla u = f(x,y) = 10, & (x,y) \in \Omega, \\ u(x,y) = 0, & (x,y) \in \partial\Omega, \end{cases} \tag{1.17}$$

where $\alpha = 0, 1$ and $\Omega = [-2,2] \times [-2,2]$. This diffusion parameter is $\mathbf{O}(1)$ except for a narrow region around the line $y = -1$, where it dips rapidly to .001. When $\alpha = 0$ there is no convection and when $\alpha = 1$, there is strong convection. We plot a solution with no convection in Fig. 1.11.

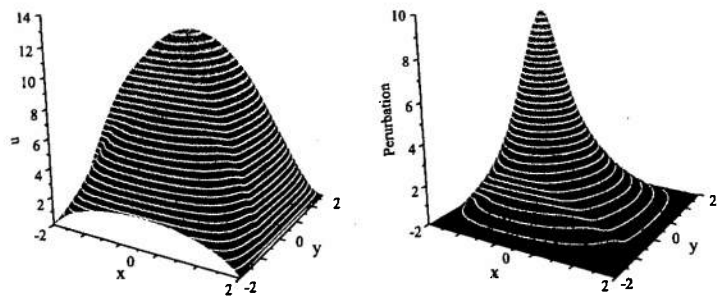


FIG. 1.11. Left: A plot of the solution of (1.17) with $\alpha = 0$. Right: A plot of the effect of the perturbation ρ when $\alpha = 0$.

We now consider the effect of perturbing the data f to $f + \rho$ by a very "pointed" function

$$\rho(x,y) = 100 \times e^{(-10 \times ((x+1)^2 + (y-.5)^2))},$$

which is nearly zero outside of a small neighborhood of $(-1, .5)$. Because the problem (1.17) is linear, we can compute the effects directly. Namely with $L(u) = f$ and U denoting the perturbed solution $L(U) = f + \rho$, we can compute the perturbation to the solution, $w = U - u$, directly as the solution of

$$L(w) = L(U - u) = L(U) - L(u) = \rho.$$

In Fig. 1.11, we plot the perturbation w to the solutions for $\alpha = 0$ and $\alpha = 1$.

We observe that the effect of the perturbation decreases dramatically to zero sufficiently far from the region where the perturbation is nonzero, e.g. close to the boundaries of $x = 2$ and $y = -2$, see Fig. 1.11. This kind of "decay of influence" is characteristic of the Greens function associated with Poisson's problem. The situation for more general elliptic problems is much more complex however.

For example, convection has a strong effect on the way in which the perturbation disturbs the solution. In Fig. 1.12, we plot the solution with $\alpha = 1$. Now, some of the effects of the perturbation are felt right across the domain, even very near the boundary at $y = -2$. Note also that the perturbation does not decay uniformly in all directions.

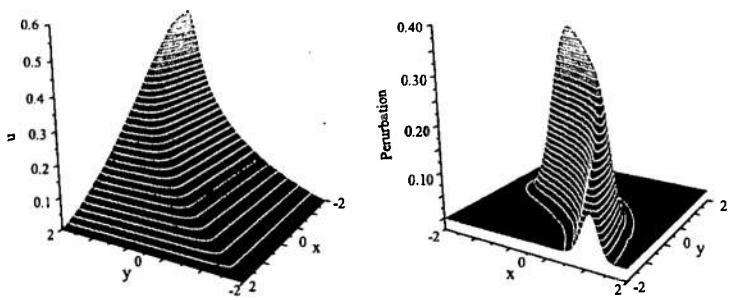


FIG. 1.12. Left: A plot of the solution of (1.17) with $\alpha = 1$. Right: A plot of the effect of the perturbation ρ when $\alpha = 1$. Note how the influence "curves" because of the singular perturbation in the diffusion.

In general, the "decay of influence" of the effects of a localized perturbation is an important stability property of elliptic problems. It has strong consequences for devising efficient adaptive mesh refinement for example. It can also be exploited to devise new approaches to domain decomposition, see (Estep *et al.*, 2005). However, the decay is very problem dependent and exploiting it fully requires numerical solution of the adjoint problem in general.

We can contrast these ideas with the classic analysis of elliptic stability, which typically yields a result of the form

$$\|w\|_* \leq C\|\rho\|_{**},$$

for some appropriate norms $\| \|_*$, $\| \|_{**}$, where ρ belongs in some reasonable space of functions and C is some constant independent of the choice of particular ρ in this space. We see that such a result does not describe the decay of influence in the example above. As with the linear algebra example, an *a priori* analysis of stability tends to be much too pessimistic when applied to a particular solution.

1.2.4 *The meaning of stability depends on the information to be computed*

In the examples above, we concentrated on the stability of pointwise values. But, we must broaden the point of view here. For example, a little reflection suggests that worrying about the pointwise behavior of Lorenz solutions is not very well motivated in terms of physical modeling. In whatever sense the Lorenz problem presents a model of weather, it is certainly not a pointwise representation of weather! Rather, it is the qualitative behavior of the Lorenz problem that is meant to represent some characteristic of weather patterns. It is more reasonable to consider a quantity of interest computed from solutions that better represents qualitative behavior of all solutions.

This is an important observation because it turns out that the effect of perturbations on a solution depends strongly on the information being computed from the solution.

Example 1.9 To illustrate this, we consider the average of the instantaneous distance from a solution of the Lorenz problem to the origin, see Fig. 1.13. The motivation is that all solutions must remain in a neighborhood of the origin. We compare results for numerical solutions with a coarse time step .001 and fine time step .0001, and see that the distances are completely different after a moderate time.

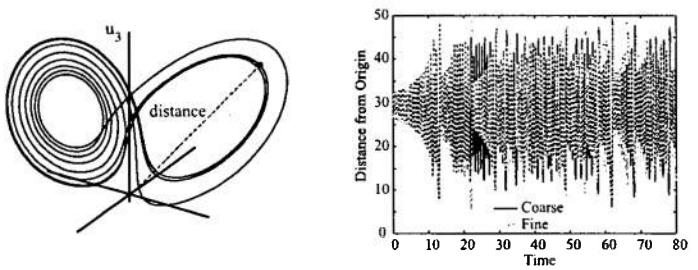


FIG. 1.13. The quantity of interest is the average of the instantaneous distance from the solution to the origin. On the right, we plot the average instantaneous distance for solutions with time steps .001 and .0001 respectively. The distances agree during an initial period and are completely different after that.

We give values for the average instantaneous distance along with the variance over three intervals in Table 1.1. The accuracy of the numerical solution appears to have little effect on the average distance.

In order to verify this observation, we compare these results to the average distance to the origin computed from an ensemble of 100 accurate solutions, each computed using time step .0001 for 15 time units in Table 1.2. Initial values for these solutions were drawn at random from values of the long time solution after $T = 50$, insuring the initial values are distributed appropriately on the strange

	Coarse Solution		Fine Solution	
End Time	Ave	Var	Ave	Var
20	27.6	52.0	27.6	51.9
80	26.5	79.5	26.5	79.2
320	26.3	83.7	26.3	83.0

TABLE 1.1. Average instantaneous distance of numerical Lorenz solutions to the origin computed with time steps .001 and .0001.

attractor. Again, there is close agreement in values.

	Coarse Solution		Fine Solution		Ensemble Average	
End Time	Ave	Var	Ave	Var	Ave	Var
320	26.3	83.7	26.3	83.0	26.3	83.7

TABLE 1.2. Average distance of numerical Lorenz solutions to the origin computed with over a long time and using an ensemble average.

Example 1.10 A characteristic of Lorenz solutions that is linked to chaotic behavior is the pattern of orbits around the nonzero fixed points. We compute the average number of orbits around a particular fixed point made by a solution before it moves to orbit the other fixed point, see Fig. 1.14. We compare

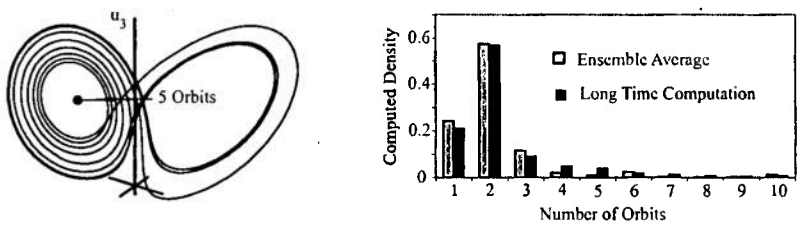


FIG. 1.14. The quantity of interest is the average number of orbits around a nonzero fixed point. On the right, we plot the probability density for the orbits computed from a long time solution and an ensemble average of short time accurate solutions.

the probability density of orbits for an ensemble average over many short time, accurate solutions to that for a long time solution.

The conclusion is that solutions of the chaotic Lorenz problem are sensitive pointwise to perturbations and errors, yet there are quantities of interest that can be computed from the solutions that are relatively insensitive to perturbation and error. An analysis to obtain accurate estimates of the effects of perturbation

and error must be conducted relative to the information that is to be computed from solutions.

1.3 The tools for quantifying stability properties: Functionals, duality, and adjoint operators

In the previous section, we saw that stability plays a critical role in determining the effects of perturbation and error. We saw that stability is often a complex issue with many facets that are not easily determined. We also saw that classic *a priori* analyses may be too crude to be used to quantify variation and error arising in particular information computed from particular solutions of particular problems with any reasonable accuracy.

All of these observations provide motivation to find another approach to determine the effects of stability. The approach we describe in this chapter is *a posteriori*, which means that the stability of particular information of a particular solution is determined after the information is computed. *A posteriori* and *a priori* analyses are fundamentally different. For example, an *a priori* error analysis of a numerical method describes the general accuracy properties for a wide class of solutions, yet generally overestimates the error in any particular solution to a significant extent. An *a posteriori* error analysis provides an estimate of the error in particular computed information, but the estimate changes when the solution changes and consequently it is generally difficult to draw any conclusions about convergence of the method. Both *a priori* and *a posteriori* analyses play key roles in analyzing the effects of uncertainty and error.

We use duality and adjoint operators to quantify stability *a posteriori*. We combine these with variational analysis to produce accurate estimates of the effects of perturbation and error. These tools have a long history of application in model sensitivity analysis and optimization, dating back to Lagrange. We present a very brief overview here, see the references (Marchuk *et al.*, 1996; Lanczos, 1997; Cacuci, 1997; Marchuk, 1995; Atkinson and Han, 2001; Aubin, 2000; Cheney, 2000; Folland, 1999; Schechter, 2002) for more details. The application of these tools to *a posteriori* error estimation has a more recent history, see the references in (Eriksson *et al.*, 1996; Eriksson *et al.*, 1995; Estep *et al.*, 2000; Becker and Rannacher, 2001; Giles and Süli, 2002; Bangerth and Rannacher, 2003; Paraschivoiu *et al.*, 1997; Barth, 2004).

1.3.1 *Functionals and computing information*

We focus on computing a particular piece of information, or a quantity of interest, from a solution of a model. We use linear functionals, which are a special kind of linear map, to do this. A continuous linear functional ℓ is a continuous linear map from a vector space X to the reals $\mathbb{R}$.

Example 1.11 Let v in $\mathbb{R}^n$ be fixed. The map $\ell(x) = v \cdot x = (x, v)$ is a linear functional on $\mathbb{R}^n$.

Example 1.12 Consider $C([a, b])$. Both $\ell(f) = \int_a^b f(x)\,dx$ and $\ell(f) = f(y)$ for $a \leq y \leq b$ are linear functionals.

Example 1.13 There are important nonlinear functionals, e.g. norms.

It is useful to think of a linear functional as providing a "one dimensional snapshot" of a vector.

Example 1.14 In Example 1.11, consider $v = e_i$, the i^{th} standard basis function. Then $\ell(x) = x_i$ where $x = (x_1, \cdots, x_n)$.

Example 1.15 If δ_y denotes the delta function at a point y in a region Ω. This gives a linear functional on sufficiently smooth, real valued functions via

$$\ell(u) = u(y) = \int_\Omega \delta_y(x) u(x)\, dx.$$

Example 1.16 The expected value $E(X)$ of a random variable X is a linear functional.

Example 1.17 The Fourier coefficients of a continuous function f on $[0, 2\pi]$,

$$c_j = \int_0^{2\pi} f(x)\, e^{-ijx}\, dx$$

Using linear functionals means settling for a set of snapshots rather than the entire solution. Presumably, it is easier to compute accurate snapshots than solutions that are accurate everywhere. In many situations, we settle for an "incomplete" set of samples.

Example 1.18 We are often happy with a small set of moments of a random variable.

Example 1.19 In applications of Fourier series, we typically use a finite sum truncation of the infinite series. We require increasing amounts of information, e.g. values, of a function in order to compute increasingly higher order Fourier coefficients.

We define the dual space to be the collection of "reasonable" snapshots. More precisely, if X is a normed vector space with norm $\| \ \|$, the space of continuous linear functionals on X is called the dual space of or on or to X, and is denoted by X^*. The dual space is a vector space. We can define the dual norm for $y \in X^*$ by

$$\|y\|_{X^*} = \sup_{\substack{x \in X \\ \|x\|_X = 1}} |y(x)| = \sup_{\substack{x \in X \\ x \neq 0}} \frac{|y(x)|}{\|x\|}.$$

Example 1.20 When $X = \mathbb{R}^n$ with the usual dot product $(\ ,\ )$ and norm $\| \ \| = \| \ \|$, we saw that every vector v in $\mathbb{R}^n$ is associated with a linear functional $\ell_v(\cdot) = (\cdot, v)$. This functional is continuous since $|(x, v)| \leq \|v\| \, \|x\|$ (The "C" in the definition is $\|v\|$). A classic result in linear algebra is that **all** linear functionals on $\mathbb{R}^n$ have this form, i.e., we can make the identification $(\mathbb{R}^n)^* \equiv \mathbb{R}^n$.

Example 1.21 Recall Hölder's inequality for $f \in L^p(\Omega)$ and $g \in L^q(\Omega)$ with $\frac{1}{p} + \frac{1}{q} = 1$ for $1 \le p, q \le \infty$ is

$$\|fg\|_{L^1(\Omega)} \le \|f\|_{L^p(\Omega)} \|g\|_{L^q(\Omega)}.$$

This implies that each g in $L^q(\Omega)$ is associated with a bounded linear functional on $L^p(\Omega)$ when $\frac{1}{p} + \frac{1}{q} = 1$ and $1 \le p, q \le \infty$ by

$$\ell(f) = \int_\Omega g(x) f(x)\, dx.$$

An important, and difficult, result is that we can "identify" $(L^p)^*$ with L^q when $1 < p, q < \infty$. The cases $p = 1, q = \infty$ and $p = \infty, q = 1$ are trickier. The case L^2 is special in that we can identify $(L^2)^*$ with L^2.

There is a useful notation for the value of a functional. If x is in X and y is in X^*, we define the bracket of x and y as

$$y(x) = \langle x, y \rangle.$$

It is not surprising that norms on X and its dual X^* are closely related. An important inequality is

Theorem 1.22 *The generalized Cauchy inequality is*

$$|\langle x, y \rangle| \le \|x\|_X \, \|y\|_{X^*}, \quad x \in X,\ y \in X^*.$$

Combining this with the Hahn-Banach theorem yields a "weak" representation of the norm on X.

Theorem 1.23 *If X is a Banach space, then*

$$\|x\|_X = \sup_{\substack{y \in X^* \\ y \neq 0}} \frac{|y(x)|}{\|y\|_{X^*}} = \sup_{\substack{y \in X^* \\ \|y\|_{X^*} = 1}} |y(x)|$$

for all x in X.

This says that we can determine the size of a vector in X by examining a sufficient number of "snapshots".

Example 1.24 In Ex. 1.20, we saw that $\mathbb{R}^n$ with the standard Euclidean norm can be identified with its dual space. Likewise, L^2 can be identified with its dual space. Both of these spaces are Hilbert spaces.

Remarkably, Ex. 1.20 generalizes to infinite dimensions. If X is a Hilbert space with inner product (x, y), then each $y \in X$ determines a linear functional $\ell_y(x) = \langle x, y \rangle = (x, y)$ for x in X. This functional is continuous by Cauchy's inequality, which says that $|(x, y)| \le \|x\| \, \|y\|$.

It turns out that is the only kind of continuous linear functional on a Hilbert space.

Theorem 1.25 Riesz Representation for Hilbert Spaces *For every bounded linear functional ℓ on a Hilbert space X, there is a unique element y in X such that*

$$\ell(x) = (x, y) \text{ for all } x \in X, \text{ and } \|y\|_{X^*} = \sup_{\substack{x \in X \\ x \neq 0}} \frac{|\ell(x)|}{\|x\|}.$$

This means that the dual space to a Hilbert space X can be identified with X. Abusing notation, it is common to replace the bracket notation and the generalized Cauchy inequality by the inner product and the "real" Cauchy inequality without comment.

1.3.2 *The adjoint operator*

To motivate the definition of the adjoint operator, let X and Y be two Banach spaces, $L : X \to Y$ be a continuous linear map, and consider the problem of computing a functional value

$$\ell\big(L(x)\big)$$

for some input $x \in X$. Some important questions are

- Given that we only want a functional value of the solution, can we find a way to compute the functional value efficiently?
- What is the error in the functional value if approximations are involved?
- Given a functional value, what can we say about x?
- Given a collection of functional values, what can we say about L?

We can address these questions using the adjoint operator. Suppose L is a continuous linear transformation. For each $y^* \in Y^*$,

$$y^* \circ L(x) = y^*(L(x)) = < Lx, y^* >$$

assigns a number to each $x \in X$, hence defines a functional $\ell(x)$. The functional $\ell(x)$ is clearly linear. It is also continuous since

$$|\ell(x)| = |y^*(L(x))| \leq \|y^*\|_{Y^*} \|L(x)\|_Y \leq \|y^*\|_{Y^*} \|L\| \, \|x\|_X = C\|x\|_X,$$

where $C = \|y^*\|_{Y^*} \|L\|$. By the definition of the dual space, there is an $x^* \in X^*$ such that $y^*(L(x)) = x^*(x)$ for all $x \in X$. x^* is unique.

We have defined $\ell = x^*$ implicitly. Given x, we first apply the operator L, then compute a functional of the result. This may seem a little strange. We put it into context of a classic example.

Example 1.26 Consider the elliptic problem

$$\begin{cases} -\Delta u = f, & y \in \Omega, \\ u = 0, & y \in \partial\Omega, \end{cases} \tag{1.18}$$

where we wish to evaluate $u(y_0)$ for some $y_0 \in \Omega$. In this example, the data f plays the role of x above in the definition of ℓ. Note that we do **not** evaluate

$f(y_0)$. Instead, we have to solve (1.18), where $u = L(f)$ is determined by the solution operator L of the Dirichlet problem. We then apply the functional

$$u(y_0) = (u, \delta_{y_0}) = (L(f), \delta_{y_0}).$$

We now apply this implicit definition to each $y^* \in Y^*$. For each y^*, assign a unique $x^* \in X^*$ and in this way define a linear transformation $L^* : Y^* \to X^*$, called the adjoint or dual operator to L.

Example 1.27 Continuing Ex. 1.26, we pose the adjoint problem

$$\begin{cases} -\Delta\phi = \delta_{y_0}, & y \in \Omega, \\ \phi = 0, & y \in \partial\Omega, \end{cases}$$

and denote the solution $\phi = L^*(\delta_{y_0})$. We have

$$(u, \delta_{y_0}) = (L(f), \delta_{y_0}) = (f, L^*(\delta_{y_0})) = (f, \phi).$$

This is just the method of Greens function.

Note that we have defined the adjoint transformation via computing snapshots using elements in the dual space. We can write these relations as

$$y^*(L(x)) = L^* y^*(x)$$

or using the bracket notation,

$$\langle L(x), y^* \rangle = \langle x, L^*(y^*) \rangle \quad x \in X,\ y^* \in Y^*. \tag{1.19}$$

Equation (1.19) is called the bilinear identity.

Example 1.28 Let $X = \mathbb{R}^m$ and $Y = \mathbb{R}^n$, where we take the standard inner product and norm. By the Riesz Representation theorem, the bilinear identity for $L \in \mathcal{L}(\mathbb{R}^m, \mathbb{R}^n)$ reads

$$(Lx, y) = (x, L^* y), \quad x \in \mathbb{R}^m,\ y \in \mathbb{R}^n.$$

We know that L is represented by a unique $n \times m$ matrix $\mathbf{A}$ so that if $y = L(x)$,

$$\mathbf{A} = \begin{pmatrix} a_{11} & \cdots & a_{1m} \\ \vdots & & \vdots \\ a_{n1} & \cdots & a_{nm} \end{pmatrix}, \quad y = \begin{pmatrix} y_1 \\ \vdots \\ y_n \end{pmatrix}, \quad x = \begin{pmatrix} x_1 \\ \vdots \\ x_m \end{pmatrix}$$

then

$$y_i = \sum_{j=1}^{m} a_{ij} x_j, \quad 1 \le i \le n.$$

For a linear functional $y^* = (y_1^*, \cdots, y_n^*)^\top \in Y^*$, we have

$$L^*y^*(x) = y^*(L(x)) = \left((y_1^*, \cdots, y_n^*), \begin{pmatrix} \sum_{j=1}^m a_{1j}x_j \\ \vdots \\ \sum_{j=1}^m a_{nj}x_j \end{pmatrix} \right) = \sum_{j=1}^m (\sum_{i=1}^n y_i^* a_{ij}) x_j$$

Therefore, $L^*(y^*)$ is given by the inner product with $\tilde{y} = (\tilde{y}_1, \cdots, \tilde{y}_m)^\mathsf{T}$ where

$$\tilde{y}_j = \sum_{i=1}^n y_i^* a_{ij}.$$

This implies the matrix $\mathbf{A}^*$ of L^* is

$$\mathbf{A}^* = \begin{pmatrix} a_{11}^* & \cdots & a_{1n}^* \\ \vdots & & \vdots \\ a_{m1}^* & \cdots & a_{mn}^* \end{pmatrix} = \begin{pmatrix} a_{11} & a_{21} & \cdots & a_{n1} \\ \vdots & & & \vdots \\ a_{1m} & a_{2m} & \cdots & a_{nm} \end{pmatrix} = A^\mathsf{T}.$$

We can write the bilinear identity as

$$y^\mathsf{T}\mathbf{A}x = x^\mathsf{T}\mathbf{A}^\mathsf{T}y$$

using the fact that $(x, y) = (y, x)$.

We give more examples below.

We conclude with some basic facts:

Theorem 1.29 *Let X and Y be normed vector spaces and $L : X \to Y$. Then, L^* is a continuous linear operator and $\|L^*\| = \|L\|$. Also, $0^* = 0$. If $M : X \to Y$ is another continuous linear operator, then $(L+M)^* = L^* + M^*$ and $(\alpha L)^* = \alpha L^*$ for all scalars α.*

If Z is another normed vector space and $N : Y \to Z$, then $NL : X \to Z$ is a continuous linear operator and $(NL)^* = L^*N^*$.

1.3.3 *Four good reasons to use adjoints*

We can provide some immediate motivations to introduce the concepts of duality adjoint operators, which indeed turn out to be fundamentally important for the analysis of operators.

Reason # 1 X^* often has good properties that X may lack.

Theorem 1.30 *If X is a normed vector space over $\mathbb{R}$, then X^* is a Banach space, i.e. Cauchy sequences in X converge to a limit in X, whether or not X is a Banach space.*

Reason # 2 There is a close connection between the stability properties of an operator and its adjoint.

Theorem 1.31 *The singular values of a matrix $\mathbf{L}$ are the eigenvalues of the square, symmetric transformations $\mathbf{L}^*\mathbf{L}$ or $\mathbf{L}\mathbf{L}^*$.*

This connects the condition number of a matrix $\mathbf{L}$ to $\mathbf{L}^*$.

Reason # 3 If L is a linear transformation between normed vector spaces, the solvability of $L(y) = b$ is closely related to the solvability of $L^*(\phi) = \psi$.

Theorem 1.32 *Let X and Y be normed linear spaces and $L : X \to Y$ a continuous linear transformation. A necessary condition that $L(x) = y$ has a solution is that $y^*(y) = 0$ for all continuous functionals y^* such that $L^*y^* = 0$. This is a sufficient condition if the range of L is closed in Y.*

Example 1.33 Suppose that $L : \mathbb{R}^m \to \mathbb{R}^n$ is associated with the $n \times m$ matrix $\mathbf{A}$, i.e., $L(x) = \mathbf{A}x$. The necessary and sufficient condition for the solvability of $\mathbf{A}x = b$ is that b is orthogonal to all linearly independent solutions of $\mathbf{A}^T y = 0$.

In general, all kinds of information about the solvability and deficiency of the linear system $\mathbf{A}x = b$ can be determined by considering $\mathbf{A}^*\mathbf{A}$. In the over-determined and under-determined cases, it yields a "natural" definition of a solution or gives conditions for a solution to exist, see (Lanczos, 1997).

Reason # 4 Suppose we wish to compute a functional $\langle \cdot, \psi \rangle$ of the solution y of an inverse problem for a linear operator A and data b,

$$Ay = b.$$

We define the adjoint (inverse) problem

$$A^*\phi = \psi.$$

Then we obtain a representation of the solution,

$$\langle y, \psi \rangle = \langle y, A^*\phi \rangle = \langle Ay, \phi \rangle = \langle b, \phi \rangle.$$

Such an error representation is very useful in practice. For example, we can compute the effect of many perturbations in the data very efficiently by computing one adjoint solution and taking inner products with the perturbations.

This formal argument is essentially the entire foundation for error estimation and uncertainty quantification described in this chapter. The reader may notice that it generalizes the method of Greens functions. We discuss this further below.

1.3.4 *Adjoint operators for linear differential equations*

We briefly discuss the computation of adjoints to differential equations. On a simple level, given a differential operator L on a domain Ω, we seek to evaluate the bilinear identity,

$$\langle Lu, v^* \rangle - \langle u, L^*v^* \rangle = 0, \quad \text{all } u \in X,\ v^* \in Y^*. \tag{1.20}$$

But, there are a lot of details needed to make this a computational process, e.g. what does it mean to compute $\langle\ ,\ \rangle$!

In the common situation in which we consider functions in a Hilbert space like $L^2(\Omega)$, we attempt to replace $\langle\ ,\ \rangle$ by the inner project $(\ ,\ )$,

$$\langle Lu, v^* \rangle - \langle u, L^* v^* \rangle = (Lu, v) - (u, L^* v).$$

Even using the familiar $L^2(\Omega)$ inner product, however, computing an adjoint can be tricky. On one hand, the process for computing an adjoint is simple to state: multiply the differential equation by a test function, integrate over the entire space-time domain (which amounts to taking the L^2 inner product of the differential equation and the test function), and keep integrating by parts until all derivatives fall on the test function. The differential operator that ends up being applied to the test function "is" the adjoint operator. On the other hand, details lead to all kinds of technical difficulties. A general abstract theory is difficult to present, see (Lions and Magenes, 1972).

First of all, the definition of the adjoint of a given forward operator depends heavily on the spaces involved with the maps. On the face, the process described above only works for functions sufficiently smooth that all the integration by parts are defined. Technically, we compute the adjoint for smooth functions and then pass to a limit (a "density" argument) to the full spaces on which the operators are defined.

Second of all, integration by parts leaves behind integrals over boundaries and these have to be accounted for when defining the adjoint operator. The reason is simply that a differential operator is generally under-determined and we add boundary and initial conditions in order to get an invertible operator. Clearly, the boundary and initial conditions therefore must affect the definition of the adjoint operator.

To simplify life, we compute the adjoint in two stages. We first assume that the functions involved are smooth and have compact support inside Ω, i.e. the functions and all their derivatives vanish at the boundary. In this way, we carry out the integration by parts while ignoring boundary terms. Given a differential operator L on a domain Ω, the formal adjoint L^* is the differential operator that satisfies

$$(Lu, v) = (u, L^* v)$$

for all sufficiently smooth u and v with compact support in Ω.

Example 1.34 For

$$Lu(x) = -\frac{d}{dx}\left(a(x)\frac{d}{dx}u(x)\right) + \frac{d}{dx}(b(x)u(x))$$

on $[0, 1]$. Integration by parts neglecting boundary terms gives the formal adjoint

$$L^* v = -\frac{d}{dx}\left(a(x)\frac{d}{dx}v(x)\right) - b(x)\frac{d}{dx}(v(x)).$$

Example 1.35 A general linear second order differential operator L in $\Omega \subset \mathbb{R}^n$ can be written

$$L(u) = \sum_{i=1}^{n}\sum_{j=1}^{n} a_{ij}\frac{\partial^2 u}{\partial x_i \partial x_j} + \sum_{i=1}^{n} b_i \frac{\partial u}{\partial x_i} + cu,$$

where $\{a_{ij}\}$, $\{b_i\}$, and c are functions of $x_1, x_2, \cdots, x_n$. Then,

$$L^*(u) = \sum_{i=1}^{n}\sum_{j=1}^{n} \frac{\partial^2 (a_{ij} v)}{\partial x_i \partial x_j} - \sum_{i=1}^{n} \frac{\partial (b_i v)}{\partial x_i} + cv.$$

It can be verified directly that

$$vL(u) - uL^*(v) = \sum_{i=1}^{n} \frac{\partial p_i}{\partial x_i},$$

where

$$p_i = \sum_{j=1}^{n} \left(a_{ij} v \frac{\partial u}{\partial x_j} - u \frac{\partial (a_{ij} v)}{\partial x_j} \right) + b_i uv.$$

The expression on the right is a divergence expression and the divergence theorem yields

$$\int_\Omega (vL(u) - uL^*(v))\, dx = \int_{\partial\Omega} p \cdot n \, ds = 0,$$

where $p = (p_1, \cdots, p_n)$ and n is the outward normal in $\partial\Omega$.

Example 1.36 Let L be a differential operator of order $2p$ of the form

$$Lu = \sum_{|\alpha|,|\beta| \le p} (-1)^{|\alpha|} D^\alpha \left(a_{\alpha\beta}(x) D^\beta u \right),$$

then

$$L^* v = \sum_{|\alpha|,|\beta| \le p} (-1)^{|\alpha|} D^\alpha \left(a_{\beta\alpha}(x) D^\beta v \right),$$

and L is elliptic if and only if L^* is elliptic. Some special cases.

$$\begin{aligned} \text{grad}^* &= -\text{div} \\ \text{div}^* &= -\text{grad} \\ \text{curl}^* &= \text{curl} \end{aligned}$$

and if

$$Lu = \sum_{|\alpha| \le p} a_\alpha(x) D^\alpha u$$

then

$$L^* v = \sum_{|\alpha| \le p} (-1)^{|\alpha|} D^\alpha (a_\alpha(x) v(x)).$$

Ignoring initial as well as boundary conditions, evolution problems are treated similarly in the sense that we integrate by parts over space and time. There is an important difference however because time has a direction and the time variable for the adjoint problem runs "backwards."

Example 1.37 If we have a parabolic problem

$$Lu = u_t - \nabla \cdot (a\nabla u) + bu, \quad x \in \Omega,\, 0 < t \leq T,$$

then

$$L^*v = -v_t - \nabla \cdot (a\nabla v) + bv, \quad x \in \Omega,\, T > t \geq 0.$$

The adjoint problem is also parabolic, and not an "ill-posed" or "backwards" parabolic problem as suggested by the "–" in front of the time derivative term. This is easily seen by making the substitution $t \rightarrow s = T - t$, so that

$$L^*v = v_s - \nabla \cdot (a(T-s)\nabla v) + b(T-s)v, \quad x \in \Omega,\, 0 < s \leq T.$$

We find it convenient to use this change of variables when solving the adjoint problem in practice.

In the second stage of computing the adjoint, we remove the assumption that the functions involved in evaluating the bilinear identity have compact support. The integrations by parts that produces the formal adjoint yield additional terms involving integrals of the functions and their derivatives over the boundary of Ω. We then choose boundary conditions for the adjoint problem depending on what we want to happen with the boundary terms from evaluating the bilinear identity.

For example, the standard approach is to pose the **minimal** boundary conditions on the adjoint problem necessary to make the boundary terms that appear when evaluating the bilinear identity vanish. These are called the adjoint boundary conditions. This definition is rather vague, but it can be made completely precise, see (Lions and Magenes, 1972).

Note that for the purpose of defining the adjoint boundary conditions, the form of the boundary conditions imposed on the original operator L are important, but the values given for these conditions are not. If the boundary conditions for L are not homogeneous, we make them so for the purpose of determining the adjoint. It follows that some of the boundary terms that appear when evaluating the bilinear identity vanish because of the homogeneous boundary conditions imposed on L and the adjoint boundary conditions insure that the rest vanish.

Example 1.38 Consider Newton's equation of motion $s''(t) = f(t)$, normalized with mass 1. If we assume $s(0) = s'(0) = 0$, and $0 < t < 1$, then we have

$$s''v - sv'' = \frac{d}{dt}(vs' - sv')$$

and

$$\int_0^1 (s''v - sv'')\,dt = (vs' - sv')\Big|_0^1. \tag{1.21}$$

Now the boundary conditions imply the contributions at $t = 0$ vanish, while at $t = 1$ we have

$$v(1)s'(1) - v'(1)s(1).$$

To insure this vanishes, we must have $v(1) = v'(1) = 0$. (We cannot specify $s(1)$ or $s'(1)$ of course.) These are the adjoint boundary conditions.

Example 1.39 Since

$$\int_\Omega (u\Delta v - v\Delta u)\,dx = \int_{\partial\Omega} \left(u\frac{\partial v}{\partial n} - v\frac{\partial u}{\partial n}\right) ds,$$

the Dirichlet and Neumann boundary value problems for the Laplacian are their own adjoints.

Example 1.40 Let $\Omega \subset \mathbb{R}^2$ be bounded with a smooth boundary and let $s =$ arclength along the boundary. Consider

$$\begin{cases} -\Delta u = f, & x \in \Omega, \\ \frac{\partial u}{\partial n} + \frac{\partial u}{\partial s} = 0, & x \in \partial\Omega. \end{cases}$$

Since

$$\int_\Omega (u\Delta v - v\Delta u)\,dx = \int_{\partial\Omega} \left(u\left(\frac{\partial v}{\partial n} - \frac{\partial v}{\partial s}\right) - v\left(\frac{\partial u}{\partial n} + \frac{\partial u}{\partial s}\right)\right) ds,$$

the adjoint problem is

$$\begin{cases} -\Delta v = g, & x \in \Omega, \\ \frac{\partial v}{\partial n} - \frac{\partial v}{\partial s} = 0, & x \in \partial\Omega. \end{cases}$$

1.4 A posteriori error analysis using adjoints

We now apply functionals, adjoint operators, and variational analysis to the problem of estimating the error of a finite element solution of a partial differential equation. The analysis rests on the observation in Reason # 4 above and we begin by extending that argument to differential equations. Given a domain Ω, which could be a time interval, a space domain, or a space-time domain, we consider a problem of the form

$$\begin{cases} Lu = f, & \text{on } \Omega, \\ \text{bound. cond. and init. val.}, & \text{on } \partial\Omega, \end{cases} \tag{1.22}$$

where L is a linear differential operator and we specify the correct boundary and/or initial conditions so that (1.22) has a unique solution. We assume that the goal of solving (1.22) is to compute a quantity of interest given as a linear

functional $\ell(u) = (u, \psi)$ for some ψ. The generalized Greens function for (1.22) corresponding to ψ satisfies

$$\begin{cases} L^*\phi = \psi, & \text{on } \Omega, \\ \text{adjoint bound. cond. and init. val.}, & \text{on } \partial\Omega, \end{cases} \tag{1.23}$$

where L^* is the formal adjoint of L. There are minor variations of this definition if we pose the data ψ on the boundary of Ω rather than the interior (i.e., as boundary or initial data), see (Wildey *et al.*, 2008). We obtain the basic representation formula,

$$(u, \psi) = (u, L^*\phi) = (Lu, \phi) = (f, \phi).$$

We use this argument to derive an *a posteriori* error estimate.

Example 1.41 We begin by returning to Ex. 1.6, and estimating the error $e = X - x$ in the numerical solution X of a linear system of equations

$$Ax = b.$$

We derive an estimate of the error in a quantity of interest given by a linear functional (e, ψ), where ψ is an given vector. We introduce the generalized Greens vector solving the adjoint problem

$$A^{\mathsf{T}}\phi = \psi.$$

Arguing as above,

$$(e, \psi) = (e, A^{\mathsf{T}}\phi) = (Ae, \phi) = (R, \phi),$$

where $R = AX - b$. We obtain a representation of the error as an inner product of the computable residual and the solution of the adjoint problem. In practice, we approximate ϕ and obtain a computable estimate.

We can also derive a bound

$$|(e, \psi)| \leq \|\phi\| \, \|R\|. \tag{1.24}$$

Returning to the specific example in Ex. 1.6, we find

$$\begin{gathered} \text{estimate of the error in the quantity of interest} \approx 1.0 \times 10^{-15}, \\ \textit{a posteriori} \text{ error bound for the quantity of interest} \approx 5.4 \times 10^{-14}, \\ \text{traditional error bound for the error} \approx 3.5 \times 10^{-5}. \end{gathered}$$

The *a posteriori* estimate is very accurate. The *a posteriori* bound overestimates the error since any cancellation in the inner product (R, ϕ) is lost, but it is still much better than the traditional condition number bound.

The adjoint quantity $\|\phi\|$ is called the stability factor. It is related to the condition number of A, since

$$\left|\left(\frac{e}{\|x\|}, \psi\right)\right| \leq \text{cond}_\psi(A)\,\frac{\|R\|}{\|b\|},$$

where

$$\text{cond}_\psi(A) = \|\phi\|\,\|A\| = \|A^{-\top}\psi\|\,\|A\|$$

is a kind of "weak" condition number of A with respect to the targeted quantity of interest. If we take the supremum of $\text{cond}_\psi(A)$ over all possible ψ with norm 1, we obtain the standard condition number of A. Hence, the stability factor obtained from the generalized Greens function is a measure of the sensitivity of particular information computed from a numerical solution of the problem to computational errors.

1.4.1 *Discretization of elliptic problems*

We first consider a general second order linear elliptic boundary value problem for a scalar unknown,

$$\begin{cases} Lu = f, & x \in \Omega, \\ u = 0, & x \in \partial\Omega, \end{cases} \tag{1.25}$$

where

$$L(D,x)u = -\nabla \cdot a(x)\nabla u + b(x)\cdot\nabla u + c(x)u(x), \tag{1.26}$$

with $u : \mathbb{R}^n \to \mathbb{R}$, a is a $n \times n$ matrix function of x, b is a n-vector function of x, and c is a function of x. We assume that $\Omega \subset \mathbb{R}^n$, $n = 2, 3$, is a smooth or polygonal domain; $a = (a_{ij})$, where $a_{i,j}$ are continuous in $\overline{\Omega}$ for $1 \leq i, j \leq n$ and there is a $a_0 > 0$ such that $v^\top a v \geq a_0$ for all $v \in \mathbb{R}^n \setminus \{0\}$ and $x \in \Omega$; $b = (b_i)$ where b_i is continuous in $\overline{\Omega}$; and finally c and f are continuous in $\overline{\Omega}$.

We discretize (1.25) by applying a finite element method to the associated variational formulation:

Find $u \in H_0^1(\Omega)$ such that

$$A(u,v) = (a\nabla u, \nabla v) + (b\cdot\nabla u, v) + (cu, v) = (f, v) \text{ for all } v \in H_0^1(\Omega), \tag{1.27}$$

where $H_0^1(\Omega)$ is the subset of functions in $H^1(\Omega)$ that are zero on $\partial\Omega$ and $H^1(\Omega)$ consists of functions that together with their first derivatives are square integrable on Ω.

To construct a finite element discretization, we form a piecewise polygonal approximation of $\partial\Omega$ whose nodes lie on $\partial\Omega$ and which is contained inside Ω. This forms the boundary of a convex polygonal domain Ω_h. We let $\mathcal{T}_h$ denote a simplex triangulation of Ω_h that is locally quasi-uniform. We let h_K denote the length of the longest edge of $K \in \mathcal{T}_h$ and define the piecewise constant mesh function h by $h(x) = h_K$ for $x \in K$. We also use h to denote $\max_K h_K$. We choose a finite element solution from the space V_h of functions that are continuous on

Ω_h piecewise linear on Ω_h with respect to $\mathcal{T}_h$, zero on the boundary $\partial\Omega_h$, and finally extended to be zero in the region $\Omega \setminus \Omega_h$. With this construction, we have $V_h \subset H_0^1(\Omega)$, and for smooth functions, the error of interpolation into V_h is $\mathbf{O}(h^2)$ in $\|\ \|$, but not better. The finite element method is:

$$\text{Compute } U \in V_h \text{ such that } A(U,v) = (f,v) \text{ for all } v \in V_h. \tag{1.28}$$

In these notes, we take for granted the usual *a priori* convergence results for finite element methods and concentrate on the *a posteriori* analysis used to produce computational error estimates. In particular, by standard results, we know that U exists and converges to u as $h \to 0$.

1.4.2 *A posteriori analysis for elliptic problems*

The goal of the *a posteriori* error analysis is to estimate the error in a quantity of interest (u,ψ) computed from the finite element solution U. To do this, we use a generalized Greens function ϕ solving the adjoint problem corresponding to ψ,

$$\begin{aligned}&\text{Find } \phi \in H_0^1(\Omega) \text{ such that}\\ &A^*(v,\phi) = (\nabla v, a\nabla\phi) - (v, \operatorname{div}(b\phi)) + (v, c\phi) = (v,\psi) \text{ for all } v \in H_0^1(\Omega).\end{aligned} \tag{1.29}$$

This is just the weak form of the adjoint problem $L^*(D,x)\phi = \psi$. Extending the analysis above,

$$\begin{aligned}(e,\psi) &= (\nabla e, a\nabla\phi) - (e, \operatorname{div}(b\phi)) + (e, c\phi)\\ &= (a\nabla e, \nabla\phi) + (b\cdot\nabla e, \phi) + (ce,\phi)\\ &= (a\nabla u, \nabla\phi) + (b\cdot\nabla u, \phi) + (eu,\phi) - (a\nabla U, \nabla\phi) - (b\cdot\nabla U, \phi) - (cU,\phi)\\ &= (f,\phi) - (a\nabla U, \nabla\phi) - (b\cdot\nabla U, \phi) - (cU,\phi).\end{aligned}$$

Letting $\pi_h\phi$ denote an approximation of ϕ in V_h, using Galerkin orthogonality (1.28), we conclude

Theorem 1.42 *The error in the quantity of interest computed from the finite element solution (1.28) satisfies the error representation,*

$$(e,\psi) = (f, \phi - \pi_h\phi) - (a\nabla U, \nabla(\phi - \pi_h\phi)) - (b\cdot\nabla U, \phi - \pi_h\phi) - (cU, \phi - \pi_h\phi), \tag{1.30}$$

where the generalized Greens function ϕ satisfies the adjoint problem (1.29) corresponding to data ψ.

The most accurate *a posteriori* error estimates are obtained by using (1.30) directly as opposed to making further estimates. To use the estimate, we approximate ϕ using a finite element method. Since $\phi - \pi_h\phi \sim \sum_{|\alpha|=2} D^\alpha\phi$ where ϕ is smooth, we use a higher order finite element than that used to solve the original boundary value problem. For example, good results are obtained using

the space V_h^2 of continuous, piecewise quadratic functions with respect to $\mathcal{T}_h$. The approximate generalized Greens function is

Compute $\Phi \in V_h^2$ such that

$$A^*(v, \Phi) = (\nabla v, a\nabla\Phi) - (v, \operatorname{div}(b\Phi)) + (v, c\Phi) = (v, \psi) \text{ for all } v \in V_h^2. \quad (1.31)$$

The approximate error representation is

$$(e, \psi) \approx (f, \Phi - \pi_h\Phi) - (a\nabla U, \nabla(\Phi - \pi_h\Phi)) - (b \cdot \nabla U, \Phi - \pi_h\Phi) - (cU, \Phi - \pi_h\Phi). \quad (1.32)$$

Example 1.43 In (Estep *et al.*, 2002), we estimate the error in the average value of the solution of

$$\begin{cases} -\Delta u = 200\sin(10\pi x)\sin(10\pi y), & (x, y) \in \Omega = [0, 1] \times [0, 1], \\ u = 0, & (x, y) \in \partial\Omega \end{cases}$$

The solution is $u = \sin(10\pi x)\sin(10\pi y)$, see Fig. 1.15.

In Fig. 1.15, we show a plot of error/estimate ratios for various degrees of accuracy. Ideally, we would get a ratio of 1. In practice, the accuracy of the estimate is affected by the numerical error in the adjoint and the errors arising from quadrature applied to the integrals in the representation (1.32). At the

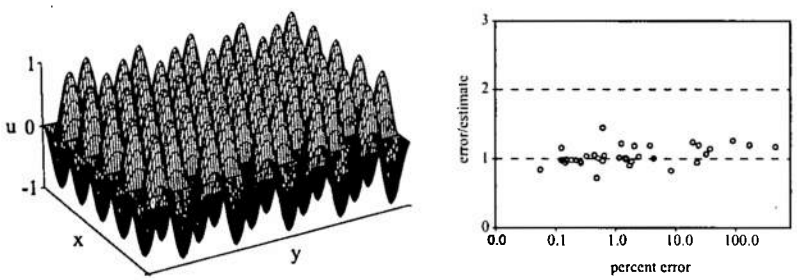


FIG. 1.15. The solution $u = \sin(10\pi x)\sin(10\pi y)$ and a plot of error/estimate ratios for various mesh sizes.

inaccurate end, we are using meshes with 5×5 to 10×10 elements. We emphasize that the computed numerical solution bears almost no resemblance to the true solution at those discretization levels, yet the estimate is reasonably accurate.

Note that in practice, the error to estimate ratio tends to vary quite a bit, even in the best of circumstances. The accuracy of the estimate is affected by numerical considerations including the accuracy of the computed adjoint solution and the use of quadrature to evaluate the integrals yielding the error estimate. In nonlinear problems, it turns out that there is a linearization error that may be significant.

1.4.3 *Adjoint analysis for nonlinear problems*

So far in the discussion of *a posteriori* analysis, we have treated linear problems. This is no coincidence because the notion of an adjoint to an operator explicitly depends on the linearity of the operator, which means in particular that the operator is independent of the input. This in turn implies that the bilinear identity serves to determine a unique adjoint operator. On reflection, this fact is apparent in the computations in Ex. 1.28.

Above, we derived an *a posteriori* error estimate for a linear problem by modifying the representation formula involving the generalized Greens vector/function. Subtracting the representation formulas for a solution and an approximation leads to a representation of the error. Now we apply the adjoint analysis technique directly to the problem of deriving an error estimate for an approximate solution of a nonlinear problem.

We assume that $F : X \to Y$ is a nonlinear map between Banach spaces X, Y with a convex domain $\mathcal{D}(F)$. Convexity is a typical assumption, with one important consequence being that mean value theorems hold. We let $u \in \mathcal{D}(F)$ solve the nonlinear problem

$$F(u) = b, \tag{1.33}$$

for some data $b \in Y$ in the range of F. We let $U \approx u$ be an approximate solution, where $U \in \mathcal{D}(F)$. The nonlinear residual of U is

$$R(U) = F(U) - b.$$

With $e = U - u$, we have

$$F(U) - F(u) = R(U). \tag{1.34}$$

Now we write $U = u + e$ and define the operator

$$\mathcal{E}(e) = \mathcal{E}(e; u) = F(u + e) - F(u),$$

where $\mathcal{E}(0) = 0$. The fundamental observation is that if F is smooth, e.g. Frechet differentiable, then $\mathcal{E}(e) \approx F'(u)e$ behaves linearly in e to first order when e is small. Hence, it makes sense to try to define an adjoint to $\mathcal{E}(e)$. The domain of $\mathcal{E}$ is

$$\mathcal{D}(\mathcal{E}) = \{v \in X | u + v \in \mathcal{D}(F)\}.$$

To be technically precise, we assume that $\mathcal{D}(\mathcal{E})$ is a dense vector subspace of X and independent of e.

We now define the adjoint operator $\mathcal{E}^*$ through

$$\langle \mathcal{E}(v), w \rangle = \langle v, \mathcal{E}(v)^* w \rangle, \quad \text{for all sufficiently small } v \in \mathcal{D}(\mathcal{E}) \text{ and all } w \in \mathcal{D}(\mathcal{E}^*). \tag{1.35}$$

(Note the notation is somewhat confusing, since $\mathcal{E}(v)$ is an operator applied to v while $\mathcal{E}(v)^*$ is an operator.) This gives the basic representation formula that is useful for error estimation. In the Sobolev space setting, we realize this as

$$(\mathcal{E}(v), w) = (v, \mathcal{E}(v)^* w), \quad \text{for all sufficiently small } v \in \mathcal{D}(\mathcal{E}) \text{ and all } w \in \mathcal{D}(\mathcal{E}^*).$$

Example 1.44 Consider the map $F : \mathbb{R}^2 \to \mathbb{R}^2$ given by

$$F(u) = \begin{pmatrix} u_1^2 + 3u_2 \\ u_1 e^{u_2} \end{pmatrix}.$$

It is easy to compute

$$\mathcal{E}(\varepsilon) = F(u+\varepsilon) - F(u) = \begin{pmatrix} 2u_1\varepsilon_1 + \varepsilon_2^2 + 3\varepsilon_2 \\ u_1 e^{u_2}\left(e^{\varepsilon_2} - 1\right) + \varepsilon_1 e^{u_2+\varepsilon_2} \end{pmatrix}.$$

We can write

$$\mathcal{E}(v) = \begin{pmatrix} 2u_1 + v_1 & 3 \\ e^{u_2+v_2} & u_1 e^{u_2}\left(\frac{e^{v_2}-1}{v_2}\right) \end{pmatrix} \begin{pmatrix} v_1 \\ v_2 \end{pmatrix}.$$

Evaluating (1.35) yields

$$\mathcal{E}(v)^* = \begin{pmatrix} 2u_1 + v_1 & e^{u_2+v_2} \\ 3 & u_1 e^{u_2}\left(\frac{e^{v_2}-1}{v_2}\right) \end{pmatrix}.$$

In the limit of small v, we recognize that $\mathcal{E}(v)^* \approx (F'(u))^*$, where $F'(u)$ is the Jacobian of F at u,

$$F'(u) = \begin{pmatrix} 2u_1 & 3 \\ e^{u_2} & u_1 e^{u_2} \end{pmatrix}.$$

This example suggests one systematic way to define an adjoint operator for error analysis. When F is Frechet differentiable, we use the Integral Mean Value Theorem to write

$$\mathcal{E}(e) = F(u+e) - F(u) = \left(\int_0^1 F'(u+se)\,ds\right) e = \left(\int_0^1 F'(su + (1-s)U)\,ds\right) e. \tag{1.36}$$

If we define the "average" Jacobian

$$\overline{F'} = \int_0^1 F'(u+se)\,ds = \int_0^1 F'(su + (1-s)U)\,ds,$$

then we can use $\mathcal{E}(e)^* = (\overline{F'})^*$ as an adjoint in the analysis.

In much of the literature on *a posteriori* error analysis for nonlinear problems, the standard way to define an adjoint operator is to use the Integral Mean Value Theorem approach. Note that in practice, u is typically unknown so $\overline{F'}$ is not computable. Typically, we simply linearize around U, i.e. replace $\overline{F'} \to F'(U)$. This may be an issue if u and U are sufficiently far apart that they are associated with significantly different adjoint operators.

Example 1.45 We continue Ex. 1.44 by examining the invertibility of $\mathcal{E}(v)^*$. We can use column operations to obtain the triangular matrix

$$\mathcal{E}(v)^* \to \begin{pmatrix} 0 & 1 \\ 3-(2u_1+v_1)u_1\left(\frac{1-\exp(-v_2)}{v_2}\right) & u_1\left(\frac{1-\exp(-v_2)}{v_2}\right) \end{pmatrix}$$

Thus, the invertibility of $\mathcal{E}(v)^*$ is determined by the distance of $3-(2u_1+v_1)u_1\left(\frac{1-\exp(-v_2)}{v_2}\right)$ to 0.

Expanding this quadratic function in u_1, we find that the roots are equal to $\pm\sqrt{3/2}$ when $v_1 = v_2 = 0$ and nearby for small v_1, v_2. If we bound u_1 away from these critical values,

$$\left|u_1^2 - \frac{3}{2}\right| \geq c^2 > 0,$$

for some constant c, we find that

$$\left|3-(2u_1+v_1)u_1\left(\frac{1-\exp(-v_2)}{v_2}\right)\right| \geq 2c^2 - |u_1|\mathbf{O}(|v_1|) - |u_1|^2\mathbf{O}(|v_2|)$$

We conclude that there is a constant $\tilde{c}$ such that $\mathcal{E}(v)^*$ is uniformly invertible for

$$|v_1| \leq \frac{\tilde{c}}{|u_1|}, \quad |v_2| \leq \frac{\tilde{c}}{|u_1|^2}.$$

Hence, linearization around two points with nearby values of u_1 produces adjoint operators with nearly the same stability (invertibility) properties.

On the other hand, if $|u_1| \approx \sqrt{3/2}$ then there are critical values of v_1 and v_2 which make the operator $\mathcal{E}(v)^*$ non-invertible. In this case, linearization around two points that are near $u_1 \approx \pm\sqrt{3/2}$ may yield adjoint operators with substantially different stability properties.

We also note that (1.35) does not define a unique adjoint operator in general.

Example 1.46 Suppose that $\mathcal{E}(e)$ can be written as $\mathcal{E}(e) = A(e)e$, where $A(e)$ is a linear operator with $\mathcal{D}(\mathcal{E}) \subset \mathcal{D}(A)$. For a fixed $e \in \mathcal{D}(\mathcal{E})$, we can define the adjoint of A satisfying $(A(e)w, v) = (w, A^*(e)v)$ for all $w \in \mathcal{D}(A)$, $v \in \mathcal{D}(A^*)$ as usual. Substituting $w = e$ shows this defines an adjoint of $\mathcal{E}$ as well. If there are several such linear operators A, then there are generally several different possible adjoints.

Following (Marchuk *et al.*, 1996), for $(t, x) \in \Omega = (0,1) \times (0,1)$, we let $X = X^* = Y = Y^* = L^2$ equal to the space of periodic functions in t and x with period 1. Consider the Burgers equation

$$u_t + uu_x + au = f, \tag{1.37}$$

where $a > 0$ is constant and f is a periodic function, and we apply periodic boundary conditions on Ω. Straightforward computation yields

$$\mathcal{E}(e) = \frac{\partial e}{\partial t} + (u+e)\frac{\partial e}{\partial x} + (u_x + a)e.$$

We have $\mathcal{E}(e) = A_1(e)e$, where

$$A_1(e)v = \frac{\partial v}{\partial t} + (u+e)\frac{\partial v}{\partial x} + (u_x + a)v$$

and the adjoint is

$$A_1(e)^* w = -\frac{\partial w}{\partial t} - \frac{\partial((u+e)w)}{\partial x} + (u_x + a)w.$$

We also have $\mathcal{E}(e) = A_2(e)e$, where

$$A_2(e)v = \frac{\partial v}{\partial t} + u\frac{\partial v}{\partial x} + (u_x + e_x + a)v$$

and

$$A_2(e)^* w = -\frac{\partial w}{\partial t} - \frac{\partial(uw)}{\partial x} + (u_x + e_x + a)w.$$

Returning to the original problem, once the adjoint is defined, we can derive a representation formula for the error. In the Sobolev space setting, we note that

$$(R(U), w) = (F(U) - F(u), w) = (\mathcal{E}(e), w) = (e, \mathcal{E}(e)^* w).$$

To estimate the error in a quantity of interest (e, ψ), we let the generalized Greens function ϕ solve

$$\mathcal{E}(e)^* \phi = \psi,$$

and we obtain

$$(e, \psi) = (R(U), \phi).$$

1.4.4 *Discretization of evolution problems*

We consider a reaction-diffusion equation for the solution u on an interval $[0, T]$,

$$\begin{cases} \dot{u} - \nabla \cdot (\epsilon(x,t)\nabla u) = f(u,x,t), & (x,t) \in \Omega \times (0,T], \\ u(x,t) = 0, & (x,t) \in \partial\Omega \times (0,T], \\ u(x,0) = u_0(x), & x \in \Omega, \end{cases} \tag{1.38}$$

where Ω is a convex polygonal domain in $\mathbb{R}^d$ with boundary $\partial\Omega$, $\dot{u}$ denotes the partial derivative of u with respect to time, and there is a constant $\epsilon > 0$ such that

$$\epsilon(x,t) \geq \epsilon, \quad x \in \Omega,\ t > 0.$$

We also assume that ϵ and f have smooth second derivatives and for simplicity, we write $f(u,x,t) = f(u)$. Everything in this paper extends directly to problems

with different boundary conditions, convection, nonlinear diffusion coefficients, and systems of equations, see (Estep *et al.*, 2000).

We describe two finite element space-time discretizations of (1.38) called the continuous and discontinuous Galerkin methods, see (Estep, 1995; Estep and French, 1994; Estep *et al.*, 2000; Estep and Stuart, 2002). We can represent many standard finite element in space - finite difference scheme in time methods as one of these two methods with an appropriate choice of quadrature for evaluating the integrals defining the finite element approximation. We partition $[0,T]$ as $0 = t_0 < t_1 < t_2 < \cdots < t_n < \cdots < t_N = T$, denoting each time interval by $I_n = (t_{n-1}, t_n]$ and time step by $k_n = t_n - t_{n-1}$. We use k to denote the piecewise constant function that is k_n on I_n. We discretize Ω using a set of elements $\mathcal{T}$ as described in Sec. 1.4.1. We describe the notation when the space mesh is the same for all time steps. In general, we can employ different meshes for each time step.

The approximations are polynomials in time and piecewise polynomials in space on each space-time "slab" $S_n = \Omega \times I_n$. In space, we let $V \subset H_0^1(\Omega)$ denote the space of piecewise linear continuous functions defined on $\mathcal{T}$, where each function is zero on $\partial\Omega$. Then on each slab, we define

$$W_n^q = \left\{ w(x,t) : \, w(x,t) = \sum_{j=0}^{q} t^j v_j(x), \, v_j \in V, \, (x,t) \in S_n \right\}.$$

Finally, we let W^q denote the space of functions defined on the space-time domain $\Omega \times [0,T]$ such that $v|_{S_n} \in W_n^q$ for $n \geq 1$. Note that functions in W^q may be discontinuous across the discrete time levels and we denote the jump across t_n by $[w]_n = w_n^+ - w_n^-$ where $w_n^\pm = \lim_{s \to t_n \pm} w(s)$.

We use a projection operator into V, $Pv \in V$, e.g. the L^2 projection satisfying $(Pv, w) = (v, w)$ for all $w \in V$, where $(\cdot, \cdot)$ denotes the $L_2(\Omega)$ inner product. We use the $\|\ \ \|$ for the L_2 norm. We also use a projection operator into the piecewise polynomial functions in time, denoted by $\pi_n : L^2(I_n) \to \mathcal{P}^q(I_n)$, where $\mathcal{P}^q(I_n)$ is the space of polynomials of degree q or less defined on I_n. The global projection operator π is defined by setting $\pi = \pi_n$ on S_n.

The continuous Galerkin cG(q) approximation $U \in W^q$ satisfies $U_0^- = P_0 u_0$ and

$$\begin{cases} \displaystyle\int_{t_{n-1}}^{t_n} \left((\dot{U}, v) + (\epsilon \nabla U, \nabla v) \right) dt = \int_{t_{n-1}}^{t_n} (f(U), v)\, dt \\ \qquad\qquad\qquad\qquad\qquad\qquad \text{for all } v \in W_n^{q-1}, \quad 1 \leq n \leq N, \\ U_{n-1}^+ = U_{n-1}^-. \end{cases} \tag{1.39}$$

Note that U is continuous across time nodes.

The discontinuous Galerkin dG(q) approximation $U \in W^q$ satisfies $U_0^- = Pu_0$ and

$$\int_{t_{n-1}}^{t_n} \left((\dot{U}, v) + (\epsilon \nabla U, \nabla v) \right) dt + \left([U]_{n-1}, v^+ \right) = \int_{t_{n-1}}^{t_n} (f(U), v)\, dt$$
$$\text{for all } v \in W_n^q, \quad 1 \le n \le N. \quad (1.40)$$

Note that the true solution satisfies both (1.39) and (1.40).

Example 1.47 To illustrate, we discretize the scalar problem

$$\begin{cases} \dot{u} - \Delta u = f(u), & (x,t) \in \Omega \times \mathbb{R}^+, \\ u(x,t) = 0, & (x,t) \in \partial\Omega \times \mathbb{R}^+, \\ u(x,0) = u_0(x), & x \in \Omega, \end{cases} \quad (1.41)$$

using the dG(0) method. Since U is constant in time on each time interval, we let $\vec{U}_n^-$ denote the M vector of nodal values with respect to the nodal basis $\{\eta_i\}_{i=1}^M$ for V. We let $\mathbf{B}: \left(B\right)_{ij} = \left(\eta_i, \eta_j\right)$ for $1 \le i,j \le M$ denote the mass matrix and $\mathbf{A}: \left(A\right)_{ij} = \left(\nabla\eta_i, \nabla\eta_j\right)$ denote the stiffness matrix. Then U_n satisfies

$$\left(\mathbf{B} + k_n\mathbf{A}\right)\vec{U}_n^- - \vec{F}(\vec{U}_n^-)k_n = \mathbf{B}\vec{U}_{n-1}^-, \quad 1 \le n \le N,$$

where $(\vec{F}(U_n^-))_i = (f(U_n^-), \eta_i)$.

As mentioned, with an appropriate use of quadrature to evaluate the integrals in the variational formulation, these Galerkin methods yield standard difference schemes. We write these standard numerical methods as space-time finite element methods in order to make use of adjoints and variational analysis.

Example 1.48 In the example above, if the lumped mass quadrature is used to evaluate the coefficients of $\mathbf{B}$, then the resulting set of equations for the dG(0) approximation is the same as the equations for the nodal values of the backward Euler - second order centered difference scheme for (1.41).

The dG(0) method is related to the backward Euler method, the cG(1) method is related to the Crank-Nicolson scheme, and the dG(1) method is related to the third order sub-diagonal Padé difference scheme, see (Jamet, 1978; Delfour and Dubeau, 1986; Delfour *et al.*, 1981; Thomée, 1980; Eriksson *et al.*, 1985; Estep and Larsson, 1993; Estep, 1995; Estep and French, 1994; Estep and Stuart, 2002).

Under general assumptions, the cG(q) and dG(q) have order of accuracy $q+1$ in time and 2 in space at any point. In addition, they enjoy a superconvergence property in time at time nodes. The dG(q) method has order of accuracy $2q+1$ in time and the cG(q) method has order $2q$ in time at time nodes for sufficiently smooth solutions.

1.4.5 *Analysis for discretizations of evolution problems*

We begin the *a posteriori* analysis by defining a suitable adjoint problem for error analysis. The adjoint problem is a linear parabolic problem with coefficients obtained by linearization around an average of the true and approximate solutions.

$$\bar{f} = \bar{f}(u, U) = \int_0^1 \frac{\partial f}{\partial u}(us + U(1-s))\, ds. \tag{1.42}$$

The regularity of u and U typically imply that $\bar{f}$ is piecewise continuous with respect to t and a continuous, H^1 function in space.

Written out pointwise for convenience, the adjoint problem to (1.38) for the generalized Greens function associated to the data ψ, which determines the quantity of interest,

$$\int_0^T (u, \psi)\, dt, \tag{1.43}$$

is

$$\begin{cases} -\dot{\phi} - \nabla \cdot (\epsilon \nabla \phi) - \bar{f}\phi = \psi, & (x,t) \in \Omega \times (T, 0], \\ \phi(x,t) = 0, & (x,t) \in \partial\Omega \times (T, 0], \\ \phi(x,T) = 0, & x \in \Omega, \end{cases} \tag{1.44}$$

Using this definition, for the dG method we have

$$\begin{aligned} \int_0^T (e, \psi)\, dt &= \int_0^T (e, -\dot{\phi} - \nabla \cdot (\epsilon \nabla \phi) - \overline{f}\phi)\, dt \\ &= \sum_{n=1}^N \int_{I_n} (e, -\dot{\phi} - \nabla \cdot (\epsilon \nabla \phi) - \overline{f}\phi)\, dt. \end{aligned}$$

We integrate by parts in time for

$$\int_{I_n} (e, -\dot{\phi})\, dt = -(e_n^-, \phi_n) + (e_{n-1}^+, \phi_{n-1}) + \int_{I_n} (\dot{e}, \phi)\, dt.$$

Likewise,

$$\int_{I_n} (e, -\nabla \cdot (\epsilon \nabla \phi))\, dt = \int_{I_n} (\epsilon \nabla e, \nabla \phi)\, dt.$$

Finally,

$$\int_{I_n} (e, \overline{f}\phi)\, dt = \int_{I_n} (\overline{f}e, \phi)\, dt = \int_{I_n} (f(U) - f(u), \phi)\, dt.$$

Next we realize that the true solution satisfies the weak formulation

$$\int_{I_n} \big((\dot{u}, \phi) + (\epsilon \nabla u, \nabla \phi) - (f(u), \phi)\big)\, dt = 0,$$

hence,

$$\begin{aligned} \int_{I_n} \big((\dot{e}, \phi) + (\epsilon \nabla e, \nabla \phi) - (f(U) - f(u), \phi)\big)\, dt \\ = \int_{I_n} \big((\dot{U}, \phi) + (\epsilon \nabla U, \nabla \phi) - (f(U), \phi)\big)\, dt, \end{aligned}$$

The sum of terms arising from the integration by parts in time simplifies

$$\sum_{n=1}^{N} \big(-(e_n^-, \phi_n) + (e_{n-1}^+, \phi_{n-1}) \big) = (e_0^+, \phi_0) + \sum_{n=1}^{N-1} (e_n^+ - e_n^-, \phi_n) - (e_N^-, \phi_N),$$

and then simplifies further upon realizing that $u_n^- = u_n^+$ and $\phi_N = 0$. Using the definition (1.40) for the dG method, we obtain

Theorem 1.49

$$\int_0^T (e, \psi)\, dt = ((I-P)u_0, \phi(0)) + \sum_{j=1}^{N} \big([U]_{j-1}, (\pi P\phi - \phi)_{j-1}^+ \big)$$
$$+ \int_0^T \big((\dot{U}, \pi P\phi - \phi) + (\epsilon(U)\nabla U, \nabla(\pi P\phi - \phi)) - (f(U), \pi P\phi - \phi) \big)\, dt. \quad (1.45)$$

The initial error is $e^-(0) = (I-P)u_0$.

If instead we desire to estimate $(u(T), \psi)$, for a function ψ, then the adjoint problem is

$$\begin{cases} -\dot{\phi} - \nabla \cdot (\epsilon \nabla \phi) - \bar{f}\phi = 0, & (x,t) \in \Omega \times (T, 0], \\ \phi(x,t) = 0, & (x,t) \in \partial\Omega \times (T, 0], \\ \phi(x,T) = \psi, & x \in \Omega. \end{cases} \quad (1.46)$$

The resulting estimate is

Theorem 1.50

$$(e(T), \psi) = ((I-P)u_0, \phi(0)) + \sum_{j=1}^{N} \big([U]_{j-1}, (\pi P\phi - \phi)_{j-1}^+ \big)$$
$$+ \int_0^T \big((\dot{U}, \pi P\phi - \phi) + (\epsilon(U)\nabla U, \nabla(\pi P\phi - \phi)) - (f(U), \pi P\phi - \phi) \big)\, dt. \quad (1.47)$$

A similar argument for the cG method, say for the global quantity of interest (1.43), yields

Theorem 1.51

$$\int_0^T (e, \psi)\, dt = ((I-P)u_0, \phi(0))$$
$$+ \int_0^T \big((\dot{U}, \pi P\phi - \phi) + (\epsilon(U)\nabla U, \nabla(\pi P\phi - \phi)) - (f(U), \pi P\phi - \phi) \big)\, dt. \quad (1.48)$$

In practice, we compute a numerical solution of a linear adjoint problem obtained from (1.44). Typically, we linearize around the computed approximate

solution and solve using a higher order method in space and time. Without specifying the details, we denote the approximate adjoint solution by Φ. Focussing on the dG method, where application to the cG method is obvious, the approximate *a posteriori* error estimate then reads

$$\begin{aligned}\left|\int_0^T (e,\psi)\,dt\right| &\approx E(U) = E(U;\psi)\\ &= \Big|((I-P)u_0,\Phi(0)) + \sum_{j=1}^{N}\big([U]_{j-1},(\pi P\Phi-\Phi)_{j-1}^{+}\big)\\ &\quad + \int_0^T \big((\dot U,\pi P\Phi-\Phi) + (\epsilon(U)\nabla U,\nabla(\pi P\Phi-\Phi)) - (f(U),\pi P\Phi-\Phi)\big)\,dt\Big|.\end{aligned} \tag{1.49}$$

Example 1.52 In (Sandelin, 2006), we consider the accuracy of the *a posteriori* error estimate applied to the chaotic Lorenz problem

$$\begin{cases}\dot u_1 = -10u_1 + 10u_2,\\ \dot u_2 = 28u_1 - u_2 - u_1u_3, & 0 < t,\\ \dot u_3 = -\frac{8}{3}u_3 + u_1u_2,\\ u_1(0) = -6.9742, u_2(0) = -7.008, u_3(0) = 25.1377.\end{cases}$$

The terms in (1.49) describing space discretization simply drop out in this case, and we compute the resulting estimate.

In Fig. 1.16, we show the accuracy of the *a posteriori* error estimate for pointwise values of each component at many times. Similar accuracy is obtained

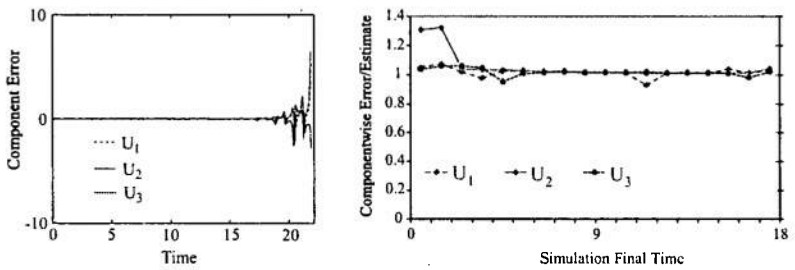


FIG. 1.16. Left: We plot an approximate error in each component of a numerical solution of the Lorenz problem computed by taking the difference between solutions with estimated error .001 and .0001 at many times. Right: we plot the pointwise error/estimate ratios for each component versus time at many time points.

for other functionals, e.g. average error.

We illustrate the idea that the solution of the adjoint problem provides a kind of condition number for the computed solution. Following (Estep and Johnson, 1998), in Fig. 1.17, we show that the adjoint solution grows very rapidly when the solution passes through the tiny region near separatix. On the other hand, the residual error of the solution remains small in this region. In this case, using only the residual, or indeed the "local error", fails completely to indicate that the error of the solution increases rapidly in a neighborhood of the separatix.

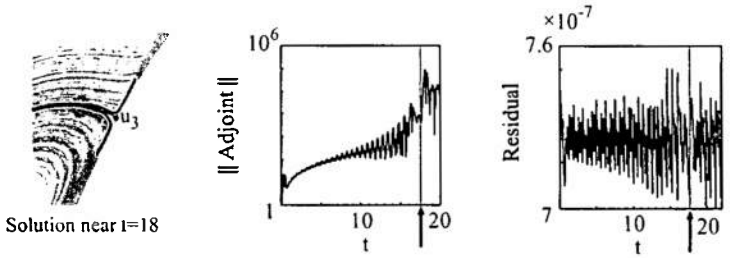


FIG. 1.17. Left: We plot the accurate and inaccurate numerical solutions during the time that the inaccurate solution becomes 100% inaccurate along with the separatix. The inaccurate solution steps on the wrong side of the separatix. Middle: We plot the norm of the adjoint solutions corresponding to pointwise error. The adjoint solution grows exponentially rapidly only when the solution passes near the separatix. Right: We plot the residual for the inaccurate solution, which remains small even when the error becomes large.

Example 1.53 In (Estep *et al.*, 2002; Estep and Williams, 1996; Estep *et al.*, 2000), we compute the *a posteriori* error estimate for the well-known bistable (Allen-Cahn) problem $\dot{u} - \epsilon\Delta u = u - u^3$ posed with Neumann boundary conditions. This is used to model the motion of domain walls in a ferromagnetic material. The problem has two attracting steady state solutions 1 and -1. Generic solutions eventually converge to one of these two steady state solutions, but the evolution towards a steady state can take some time because of the interesting competition between competing stable processes, e.g. the diffusion that tends to drive a solution towards zero and the reaction that tends to drive a solution towards 1 or -1.

In one dimension, generic solutions form a pattern of layers between the values of -1 and 1, then solutions undergo long periods of metastability during which the motion of the layers "horizontally" is extremely slow punctuated by rapid transients in which the solution moves to another metastable state or the final stable state. We show the evolution of numerical approximation of a metastable solution with two metastable periods $[0, 44]$ and $[44, 144]$ in Fig. 1.18. The timescale of metastable periods increases exponentially in $1/\sqrt{\epsilon}$ as the diffusion coefficient

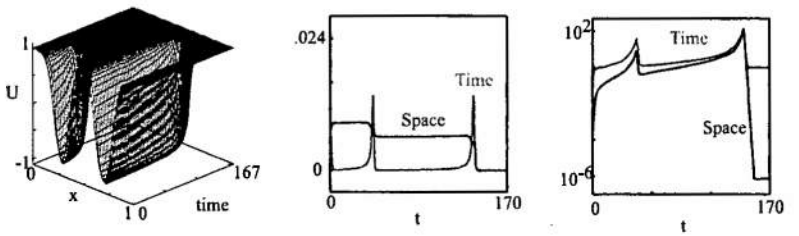


FIG. 1.18. Left: The evolution of a metastable solution of the bistable problem with $\epsilon = .0009$. Middle: Evolution of the time and space residuals on a uniform discretization. Right: The evolution of the absolute adjoint weights for pointwise errors for time and space.

ϵ decreases.

We plot the L^2 space norms of the residuals versus time for numerical solutions computed on uniform discretizations. The time residuals reflect an initial transient and then the two transients concluding the metastable periods. The space residual simply becomes smaller as the layers disappear. Finally, we plot the L^2 space norm of the adjoint weights corresponding to the time and space parts of the error estimate for the quantities of interest equal to pointwise values of the solution at a set of uniform time points. The weights grow in advance of the transients concluding the metastable periods but immediately decrease to 1 or smaller right after the transient, indicating that accumulated errors have damped. Thus, while the effects of errors grow during metastable periods, the overall error accumulation remains bounded, implying that accurate long time solutions can be computed provided the meshes are sufficiently refined.

In two dimensions, the dynamics of the problem are much different because the evolution is governed by "motion by mean curvature", meaning that the normal velocity of a transition layer is proportional to the sum of the principle curvatures of the layer. Consequently, the time scale for the evolution increases only at an algebraic rate, κ/ϵ, where κ is the mean curvature, as the diffusion coefficient ϵ decreases. We solve the bistable problem using initial data consisting of two "mesas" corresponding to the two wells in the solution shown in Fig. 1.18 using $\epsilon = .00003$ so that the evolution occurs over the same time scale. We show four snapshots of the solution in Fig. 1.19. The time evolution of the adjoint weights show a pattern of growth and decay as for metastable solutions in one dimension. However, solutions in two dimensions are much less sensitive to perturbations than solutions in one dimension, and the adjoint weights are much smaller overall.

1.4.6 *General comments on a posteriori analysis*

We can abstract the four steps for *a posteriori* error analysis as

1. Identify functionals that yield the quantities of interest

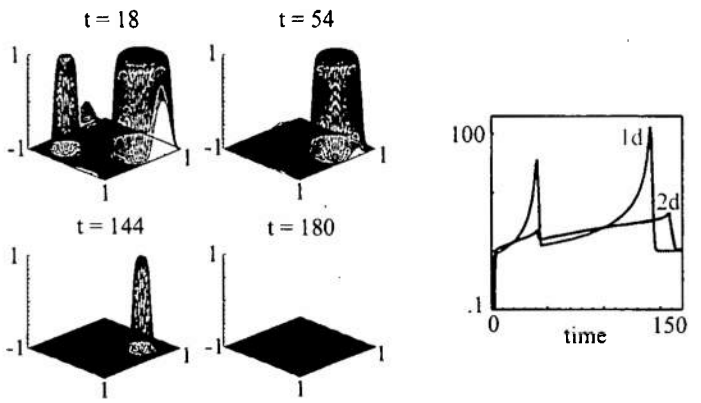


FIG. 1.19. Left: Four snapshots of a bistable solution in two dimensions. Right: The evolution of the absolute adjoint weights for pointwise errors for time and space along with a the weights for a solution in one dimension.

2. Define appropriate adjoint problems for the quantities of interest
3. Derive a computable residual for each source of error
4. Derive an error representation using a suitable adjoint weights for each residual

We also note that in general we have to account for all sources of error in the analysis. Typical sources include

- space and time discretization (approximation of the solution space)
- use of quadrature to compute integrals in a variational formulation (approximation of the differential operator)
- solution error in solving any linear and nonlinear systems of equations
- model error
- data and parameter error
- operator decomposition

We have not discussed most of these sources in this note. However, it is important to realize that different sources of error typically accumulate and propagate at different rates, and so must be accounted for individually in any analysis.

1.5 A posteriori error estimates and adaptive mesh refinement

Computing accurate error estimates provides the tantalizing idea of optimizing discretizations. We briefly discuss the use of *a posteriori* error estimates for guiding adaptive mesh refinement.

A typical goal of adaptive error control is to generate a mesh with a relatively small number of elements such that for a given tolerance TOL and data ψ,

$$\text{error in the quantity of interest } = |(e,\psi)| \lessapprox \text{TOL}. \tag{1.50}$$

We note that (1.50) cannot be verified in practice because the error is unknown, so we use an error estimate and try to construct a mesh to achieve

$$a\ \textit{posteriori}\ \text{estimate of the error in the quantity of interest } \lessapprox \text{TOL}. \tag{1.51}$$

The general idea is to write the estimate as a sum of "element contributions" that indicate the contribution from discretization on each element to the total error. We identify the elements that contribute most and then refine those elements.

However, this simple description belies a number of theoretical and practical difficulties.

1.5.1 *Adaptive mesh refinement in space*

We first consider adaptive mesh refinement for a stationary problem. In the case of an elliptic problem, we use the estimate (1.32) to implement (1.51). To do so, we rewrite (1.32) as a sum of (signed) element contributions,

$$(e,\psi) \approx \sum_{K\in\mathcal{T}_h} \int_K \big((f - b\cdot\nabla U - cU)(\Phi - \pi_h\Phi) - a\nabla U\cdot\nabla(\Phi - \pi_h\Phi)\big)\,dx. \tag{1.52}$$

Thus using (1.52), (1.51) gives the goal of satisfying the following condition: The mesh acceptance criterion is

$$\left|\sum_{K\in\mathcal{T}_h} \int_K \big((f - b\cdot\nabla U - cU)(\Phi - \pi_h\Phi) - a\nabla U\cdot\nabla(\Phi - \pi_h\Phi)\big)\,dx\right| \le \text{TOL}. \tag{1.53}$$

If the current approximation satisfies (1.53), then the solution is deemed acceptable and the refinement process is stopped.

The difficulties start when (1.53) is not satisfied. We have to decide how to "enrich" the discretization, e.g., refine the mesh or increase the order of the element functions, in order to improve the accuracy. The problem is that generally there is a great deal of cancellation among the contributions from each element. For example, consider that large positive contributions from one subregion might cancel the large negative contributions from another region so that the sum of the contributions from the two regions together is small, see Ex. 1.54 below. In fact, we make the certainly controversial claim that

There is currently no theory or practical method for accommodating cancellation of errors in an adaptive error control in a way that truly optimizes efficiency.

The standard approach is to formulate the discretization enrichment problem as a constrained optimization problem after replacing the error estimate by an

error bound consisting of a sum over elements of positive quantities. For example, we obtain a bound from (1.52) by inserting norms in some way, e.g., we use

$$|(e,\psi)| \leq \sum_{K\in\mathcal{T}_h} \int_K \left|(f - b\cdot\nabla U - cU)(\Phi - \pi_h\Phi) - a\nabla U\cdot\nabla(\Phi-\pi_h\Phi)\right| dx. \quad (1.54)$$

Thus, if (1.53) is **not** satisfied, then the mesh is refined in order to achieve the more conservative criterion,

$$\sum_{K\in\mathcal{T}_h} \int_K \left|(f - b\cdot\nabla U - cU)(\Phi - \pi_h\Phi) - a\nabla U\cdot\nabla(\Phi-\pi_h\Phi)\right| dx \lessapprox \text{TOL}. \quad (1.55)$$

The adaptive error control problem is the constrained minimization problem of finding a mesh with a minimal number of degrees of freedom on which the approximation satisfies (1.55). Using the fact that the bound in (1.55) is a sum of positive terms, and assuming the solution is asymptotically accurate, a calculus of variations argument yields the **Principle of Equidistribution**, which states that the solution of this constrained optimization problem is achieved when the elements contributions are all approximately equal. An adaptive mesh algorithm is a procedure for solving the constrained minimization problem associated with (1.55). If the Principle of Equidistribution is used, then the algorithm seeks to choose meshes so that the element contributions are approximately equal.

Depending on the argument, two possible element acceptance criterion for the element indicators are

$$\max_K \left|(f - b\cdot\nabla U - cU)(\Phi - \pi_h\Phi) - a\nabla U\cdot\nabla(\Phi-\pi_h\Phi)\right| \lesssim \frac{\text{TOL}}{|\Omega|}, \quad (1.56)$$

or

$$\int_K \left|(f - b\cdot\nabla U - cU)(\Phi - \pi_h\Phi) - a\nabla U\cdot\nabla(\Phi-\pi_h\Phi)\right| dx \lesssim \frac{\text{TOL}}{M}, \quad (1.57)$$

where M is the number of elements in $\mathcal{T}_h$. Elements that fail one of these tests are marked for refinement.

Computing a mesh using these criteria is usually performed by a "compute-estimate-mark-refine" adaptive algorithm that begins with a coarse mesh and then refines those elements on which (1.56) respectively (1.57) fail successively. See (Eriksson *et al.*, 1995; Eriksson *et al.*, 1996; Becker and Rannacher, 2001; Bangerth and Rannacher, 2003; Estep *et al.*, 2005; Carey *et al.*, 2008*b*) for more details.

The problem with any claims of "optimal" mesh selection is that generically the bound (1.54) is typically orders of magnitude larger than the estimate (1.52).

Example 1.54 We illustrate the issue of the effect of cancellation of errors on the choice of an optimal adapted mesh with a simple computation. Assume that we solve an elliptic problem on a square domain using bilinear elements on a mesh

+.001	+.001	+.001	+.01	+.01
+.001	+.001	+.001	+.01	+.01
+.001	+.001	+.001	+.001	+.001
-.0333	-.0333	+.001	+.001	+.001
+.1	-.0333	+.001	+.001	+.001

+.001	+.001	+.001	+.01	+.01
+.001	+.001	+.001	+.01	+.01
+.001	+.001	+.001	+.001	+.001
.0333	.0333	+.001	+.001	+.001
+.1	.0333	+.001	+.001	+.001

FIG. 1.20. On the left, we display (simulated) signed element contributions. We shade four elements chosen for refinement using a nonstandard criteria. On the right, we display the corresponding absolute element contributions and shade four elements marked for refinement by a standard adaptive algorithm.

consisting of rectangles and the *a posteriori* estimate yields the signed element contributions shown on the left in Fig. 1.20. The total *a posteriori* estimate is

$$.1 + 3 \times -.0333 + 17 \times .001 + 4 \times .01 = .0571.$$

Note that the large element contribution in the lower left corner is nearly canceled by the contributions of its three neighbors, $.01+3\times-.0333 = .0001$, so that region ends up contributing relatively little to the error. If we refine in the upper right hand corner by subdividing each square into four smaller elements as indicated (and assume the element contributions decrease by a factor of $2^2 = 4$ without any change in sign), the new estimate becomes

$$.1 + 3 \times -.0333 + 17 \times .001 + 16 \times .01 \times \frac{1}{4} \times \frac{1}{4} = .0271.$$

Note that while we change 4 elements into 16 smaller elements, the element contribution in each goes down by a factor of 4 while the area of each smaller element is 4 times smaller than its parent.

On the other hand, if we use the absolute element contributions to guide refinement, then the elements in the lower left hand corner are refined as shown on the left in Fig. 1.20, the new estimate becomes

$$4\times.1\times\frac{1}{4}\times\frac{1}{4}+3\times\left(4\times-.0333\times\frac{1}{4}\times\frac{1}{4}\right)+17\times.001+16\times.01\times\frac{1}{4}\times\frac{1}{4}=.057025.$$

There is almost no improvement in overall accuracy in the quantity of interest.

Regardless of the issue of dealing with cancellation of errors efficiently, there is still a crucial difference between adaptive mesh refinement based on adjoint-weighted residual estimates and traditional "error indicators" that often amount to using only residuals or "local errors." In the adjoint-based approach, the

element residuals are scaled by an adjoint weight, which reflects how much error in that element affects the error in the quantity of interest. This has a significant effect of mesh refinement patterns in general.

Example 1.55 In (Estep *et al.*, 2005), we apply these ideas to the adaptive solution of

$$\begin{cases} -\nabla \cdot \left((.05 + \tanh(10(x-5)^2 + 10(y-1)^2))\nabla u\right) \\ \qquad\qquad + \begin{pmatrix} -100 \\ 0 \end{pmatrix} \cdot \nabla u = 1, \quad (x,y) \in \Omega = [0,10] \times [0,2], \\ u = 0, \quad (x,y) \in \partial\Omega \end{cases} \tag{1.58}$$

In Fig. 1.21 we show the mesh required to obtain a numerical solution whose average value is accurate to within 4%. The adaptive pattern is obtained by refining from a coarse uniform mesh using (1.57). Convection causes a nonuniform pattern of refinement.

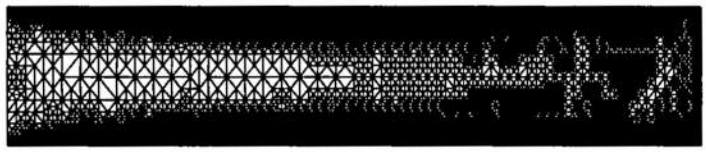


FIG. 1.21. The mesh used to solve (1.58) with an error of 4% in the average value requires 24,4000 elements.

In the first computation, the quantity of interest is given by a function ψ that is constant over the entire domain. In the next computation, we take the quantity of interest to be the average value in a square in one corner of the domain. We now require much fewer elements to achieve the desired accuracy. The pattern of refinement shows the effects of the adjoint solution, see Fig. 1.22 and Fig. 1.25. In particular, the adjoint solution decreases rapidly to zero towards the side of the domain opposite to the quantity of interest region and there is less dense mesh refinement along that side. The influence of regions far "upstream" is also diminished.

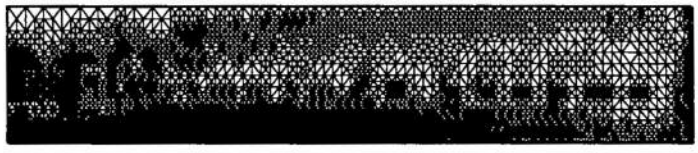


FIG. 1.22. The quantity of interest is the average value in the shaded square. The final mesh requires 7300 elements.

In the next computation, we take the quantity of interest to be the average value in a square in middle of the domain. Again, the pattern of refinement shows the effects of the adjoint solution, see Fig. 1.23 and Fig. 1.25. In particular, the

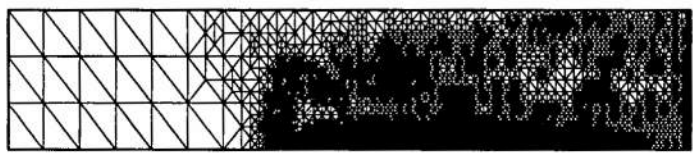


FIG. 1.23. The quantity of interest is the average value in the shaded square. The final mesh requires 7300 elements.

adaptive mesh refinement makes no attempt to resolve the boundary layer at the "outflow" boundary as the accuracy there has no effect on the accuracy of the quantity of interest.

In the final computation, we take the quantity of interest to be the average value in a square in at the far end of the domain. Again, the pattern of refinement shows the effects of the adjoint solution, see Fig. 1.24 and Fig. 1.25.

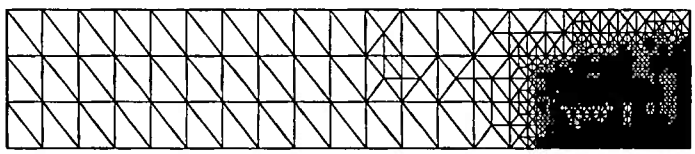


FIG. 1.24. The quantity of interest is the average value in the shaded square. The final mesh requires 3500 elements.

In Fig. 1.25, we plot the solutions of the adjoint problems corresponding to the quantities of interest equal to average values in squares at the opposite ends of the domain.

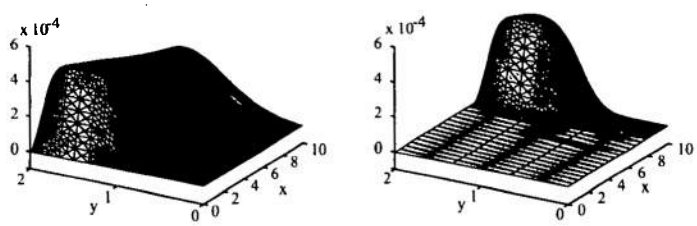


FIG. 1.25. The adjoint solutions for the computations in Fig. 1.22 and Fig. 1.23.

1.5.2 *Adaptive mesh refinement for evolutionary problems*

Traditionally, different approaches for adaptive mesh algorithms are used to handle spatial meshes and time discretization. Influenced by the long history of "local error control", the traditional time algorithm achieves an equidistribution of element contributions by insuring that the contribution from each time interval is smaller than, but approximately equal to, a "local error tolerance" LTOL before proceeding to the next time step. Often, LTOL is input directly without any attempt to relate it to the desired tolerance TOL on the error. Given a true global error estimate however and the asymptotic accuracy of the integration scheme, there are various heuristic arguments for determining LTOL in terms of TOL.

For an evolutionary partial differential equation, space and time mesh refinement strategies have to be combined somehow. In the case of a parabolic problem, we distinguish time and space contributions in (1.49) by "splitting" the projections on the adjoint solution,

$$\begin{aligned} E(U) = \Big| &((I-P)u_0, \Phi(0)) + \sum_{j=1}^{N} \big([U]_{j-1}, (P\Phi - \Phi)^+_{j-1}\big) \\ &+ \int_0^T \big((\dot{U}, P\Phi - \Phi) + (\epsilon(U)\nabla U, \nabla(P\Phi - \Phi)) - (f(U), P\Phi - \Phi)\big)\, dt \\ &+ \sum_{j=1}^{N} \big([U]_{j-1}, ((\pi - 1)P\Phi)^+_{j-1}\big) \\ &+ \int_0^T \big((\dot{U}, (\pi - 1)P\Phi) + (\epsilon(U)\nabla U, \nabla((\pi - 1)P\Phi)) \\ &\qquad - (f(U), (\pi - 1)P\Phi)\big)\, dt \Big|. \end{aligned} \tag{1.59}$$

We define bounds on the time and space contributions,

$$\begin{aligned} \text{Æ}_t(U) = &\sum_{j=1}^{N} \sum_{K \in \mathcal{T}_h} \Big| \int_K ([U]_{j-1})(((\pi - 1)P\Phi)^+_{j-1})\, dx \Big| \\ &+ \sum_{n=1}^{N} \sum_{K \in \mathcal{T}_h} \Big| \int_{t_{n-1}}^{t_n} \int_K \big(\dot{U}((\pi - 1)P\Phi) + (\epsilon(U)\nabla U) \cdot (\nabla(\pi - 1)P\Phi) \\ &\qquad - f(U)((\pi - 1)P\Phi)\big)\, dx dt \Big|, \end{aligned} \tag{1.60}$$

$$\begin{aligned}\text{Æ}_x(U) = &\sum_{K\in\mathcal{T}_h}\sum_{j=1}^{N}\sum_{K\in\mathcal{T}_h}\left|\int_K([U]_{j-1})((P\Phi-\Phi)^+_{j-1})\,dx\right| \\ &+\sum_{n=1}^{N}\sum_{K\in\mathcal{T}_h}\left|\int_{t_{n-1}}^{t_n}\int_K\big(\dot{U}(P\Phi-\Phi)+(\epsilon(U)\nabla U)\cdot(\nabla(P\Phi-\Phi))\right. \\ &\left.-f(U)(P\Phi-\Phi)\big)\,dxdt\right|. \quad (1.61)\end{aligned}$$

Note that the space discretization may affect the time contribution and likewise the time discretization may affect the space contribution.

We may now split the adaptive mesh problem into two sub-problems, refining the space and time steps in order to achieve

$$\text{Æ}_x(U) \lessapprox \frac{\text{TOL}}{2} \text{ and } \text{Æ}_t(U) \lessapprox \frac{\text{TOL}}{2}. \quad (1.62)$$

On a given time interval, this requires an iteration during which both the space mesh and time steps are refined.

1.6 Multiscale operator decomposition

We now turn to the main goal of this chapter, which is to describe how the techniques of *a posteriori* error analysis can be extended to multiscale operator decomposition solutions of multiphysics, multiscale problems. Recall that general approach is to decompose the multiphysics problem into components involving simpler physics over a relatively limited range of scales, and then to seek the solution of the entire system through some sort of iterative procedure involving solutions of the individual components.

While the particulars of the analysis vary considerably with the problem, there are several key ideas underlying a general approach to treat operator decomposition multiscale methods, including:

- We identify auxiliary quantities of interest associated with information passed between physical components and solve auxiliary adjoint problems to estimate the error in those quantities.
- We deal with scale differences by introducing projections between discrete spaces used for component solutions and estimate the effects of those projections.
- The standard linearization argument used to define an adjoint operator associated with error analysis for a nonlinear problem may fail, requiring another approach to define adjoint operators.
- In this regard, the adjoint operator associated with a multiscale operator decomposition solution method is often different than the adjoint associated with the original problem, and the difference may have a significant impact on the stability of the method.

- In practice, solving the adjoint associated with the original fully-coupled problem may present the same kinds of multiphysics, multiscale challenges posed by the original problem, so attention must be paid to the solution of the adjoint problem.

We explain these ideas in the context of three examples.

1.6.1 *Multiscale decomposition of triangular systems of elliptic problems*

Following (Carey *et al.*, 2006), we can capture the essential features of the thermal actuator model described in Example 1.1 using a two component "one-way" coupled system of the form

$$\begin{cases} -\nabla \cdot a_1 \nabla u_1 + b_1 \cdot \nabla u_1 + c_1 u_1 = f_1(x), & x \in \Omega, \\ -\nabla \cdot a_2 \nabla u_2 + b_2 \cdot \nabla u_2 + c_2 u_2 = f_2(x, u_1, Du_1), & x \in \Omega, \\ u_1 = u_2 = 0, & x \in \partial\Omega, \end{cases} \tag{1.63}$$

where a_i, b_i, c_i, f_i are smooth functions, with $a_1, a_2 \geq \alpha > 0$ on a bounded domain Ω in $\mathbb{R}^N$ with boundary $\partial\Omega$, and α is a constant. Note that the problems are coupled through f_2. The "lower triangular" form of this system means that we can either solve it as a coupled system or we can solve the first equation and then use the solution to generate the parameters for the second problem. The latter approach fits the idea of a multiscale, operator decomposition discretization.

The weak form of the first component of (1.63) reads: find $u_1 \in \tilde{W}_2^1(\Omega)$ satisfying

$$\mathcal{A}_1(u_1, v_1) = (f_1, v_1), \text{ for all } v_1 \in H_0^1(\Omega), \tag{1.64}$$

where

$$\mathcal{A}_1(u_1, v_1) \equiv (a_1 \nabla u_1, \nabla v_1) + (b_1(x) \cdot \nabla u_1, v_1) + (c_1 u_1, v_1)$$

is a bilinear form on Ω and $H_0^1(\Omega)$ is the subspace of functions in $H^1(\Omega)$ that are zero on $\partial\Omega$. Likewise the weak formulation of the second component of (1.63) reads: find $u_2 \in H_0^1(\Omega)$ satisfying

$$\mathcal{A}_2(u_2, v_2) = (f_2(x, u_1, Du_1), v_2), \text{ for all } v_2 \in H_0^1(\Omega), \tag{1.65}$$

where

$$\mathcal{A}_2(u_2, v_2) \equiv (a_2 \nabla u_2, \nabla v_2) + (b_2(x) \cdot \nabla u_2, v_2) + (c_2 u_2, v_2),$$

is another bilinear form on Ω.

We introduce the finite element space $\mathcal{S}_{h,1}(\Omega) \subset H_0^1(\Omega)$, corresponding to a discretization $\mathcal{T}_{h,1}$ of Ω for the first component, and another finite element space $\mathcal{S}_{h,2}(\Omega)$, on a different mesh $\mathcal{T}_{h,2}$, for the second component. Using different finite element spaces for different components in a system of equations raises a serious practical difficulty. Namely, evaluating integrals defining finite element approximate solutions involve functions from different spaces is problematic. In practice, quadrature formulas are used to approximate the integrals defining a finite element function. This raises a potential difficulty because quadrature

formulas work best when the integrands are smooth, whereas the standard finite element functions are **only** continuous. We avoid potential difficulties by writing any integrals as a sum of integrals over elements,

$$\int_\Omega \text{integrand}\, dx = \sum_{K \in \mathcal{T}_h} \int_K \text{integrand}\, dx,$$

and applying quadrature formulas on each element on which the finite element functions are smooth. However, in the case of a system in which the components are solved in different finite element spaces, it is not so straightforward to apply quadrature formulas to evaluate integrals. A function in one finite element space may only be continuous on an element associated with another finite element space. To avoid this problem, we introduce projections $\Pi_{i \to j}$ from $\mathcal{S}_{h,i}$ to $\mathcal{S}_{h,j}$, e.g. interpolants or an L^2 orthogonal projection. We apply these projections before applying quadrature formulas.

Algorithm 2 Multiscale Operator Decomposition for Triangular Systems of Elliptic Equations

Construct discretizations $\mathcal{T}_{h,1}, \mathcal{T}_{h,2}$ and corresponding spaces $\mathcal{S}_{h,1}, \mathcal{S}_{h,2}$
Compute $U_1 \in \mathcal{S}_{h,1}(\Omega)$ satisfying

$$\mathcal{A}_1(U_1, v_1) = (f_1, v_1), \text{ for all } v_1 \in \mathcal{S}_{h,1}(\Omega). \tag{1.66}$$

Compute $U_2 \in \mathcal{S}_{h,2}(\Omega)$ satisfying

$$\mathcal{A}_2(U_2, v_2) = (f_2(x, \Pi_{1\to 2}U_1, \Pi_{1\to 2}DU_1), v_2), \text{ for all } v_2 \in \mathcal{S}_{h,2}(\Omega). \tag{1.67}$$

We observe that any errors made in the solution of the first component affect the solution of the second component. This turns out to be a crucial fact for *a posteriori* error analysis.

Example 1.56 In (Carey *et al.*, 2006), we solve a system

$$\begin{cases} -\Delta u_1 = \sin(4\pi x)\sin(\pi y), & x \in \Omega \\ -\Delta u_2 = b \cdot \nabla u_1 = 0, & x \in \Omega, \\ u_1 = u_2 = 0, & x \in \partial\Omega, \end{cases} \qquad b = \frac{2}{\pi}\begin{pmatrix} 25\sin(4\pi x) \\ \sin(\pi x) \end{pmatrix} \tag{1.68}$$

using a standard piecewise linear, continuous finite element method, where $\Omega = [0,1] \times [0,1]$, in order to compute the quantity of interest

$$u_2(.25, .25).$$

We solve for u_1 first and then solve for u_2 using independent meshes and show the solutions in Fig. 1.26.

Using uniform meshes, evaluating the standard *a posteriori* error estimate for the second component problem, ignoring any effect arising from error in the

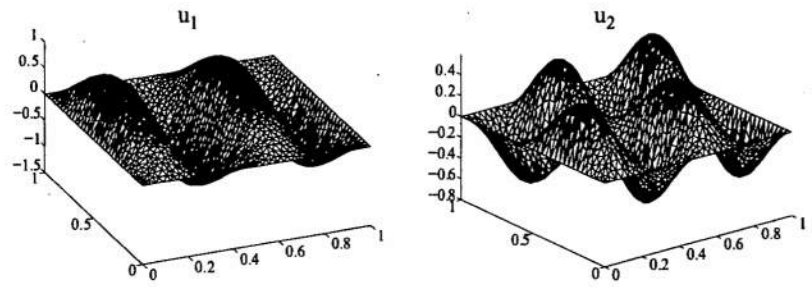


FIG. 1.26. Solutions of the component problems of (1.68) computed on uniform meshes.

solution of the first component, yields an estimate of component solution error to be $\approx .0042$. However, the true error is $\approx .0048$ and there is discrepancy of $\approx .0006$ ($\approx 13\%$) in the estimate. This is a consequence of ignoring the transfer error.

If we adapt the mesh for the solution of the second component based on the *a posteriori* error estimate of the error in that component while neglecting the effects of the decomposition, see Fig. 1.27, the discrepancy becomes alarmingly worse. For example, we can refine the mesh until the estimate of the error in the second component is $\approx .0001$. But, we find that the true error is $\approx .2244$!

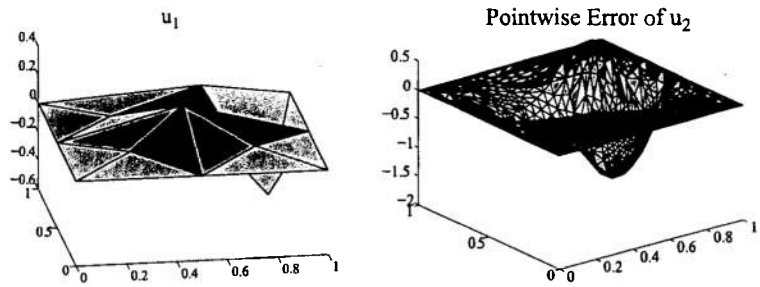


FIG. 1.27. Results obtained after refining the mesh for the second component so that the *a posteriori* error estimate of the error only in the second component is less than .001. The mesh for the first component remains coarse, consequently the error in the first component becomes relatively much larger.

1.6.1.1 *A linear algebra example* We can describe some of the essential features of the analysis using a block lower triangular linear system of equations. We consider the system

$$\begin{pmatrix} \mathbf{A}_{11} & \mathbf{0} \\ \mathbf{A}_{21} & \mathbf{A}_{22} \end{pmatrix} \begin{pmatrix} u_1 \\ u_2 \end{pmatrix} = \begin{pmatrix} b_1 \\ b_2 \end{pmatrix} = b\,, \tag{1.69}$$

with approximate solution

$$U = \begin{pmatrix} U_1 \\ U_2 \end{pmatrix} \approx \begin{pmatrix} u_1 \\ u_2 \end{pmatrix} = u.$$

We estimate the error in a primary quantity of interest involving only u_2,

$$(\psi^{(1)}, u) = (\psi_2^{(1)}, u_2) \text{ where } \psi = \begin{pmatrix} 0 \\ \psi_2^{(1)} \end{pmatrix}.$$

We require a superscript (1) since we later define an auxiliary quantity of interest.The lower triangular structure of the system matrix yields residuals

$$\begin{aligned} R_1 &= b_1 - \mathbf{A}_{11} U_1\,; \\ R_2 &= (b_2 - \mathbf{A}_{21} U_1) - \mathbf{A}_{22} U_2\,. \end{aligned}$$

Note that the residual R_2 of the approximate solution of the second component depends upon the solution of the first component, and any attempt to decrease this residual requires a consideration of the accuracy of U_1. The adjoint problem to (1.69) is

$$\begin{pmatrix} \mathbf{A}_{11}^{\mathsf{T}} & \mathbf{A}_{21}^{\mathsf{T}} \\ \mathbf{0} & \mathbf{A}_{22}^{\mathsf{T}} \end{pmatrix} \begin{pmatrix} \phi_1^{(1)} \\ \phi_2^{(1)} \end{pmatrix} = \begin{pmatrix} 0 \\ \psi_2^{(1)} \end{pmatrix},$$

and the resulting error representation is

$$\begin{aligned} (\psi^{(1)}, e) = (\psi_2^{(1)}, e_2) &= (\mathbf{A}_{22}^{\mathsf{T}} \phi_2^{(1)}, e_2) \\ &= (\phi_2^{(1)}, \mathbf{A}_{22} u_2) - (\phi_2^{(1)}, \mathbf{A}_{22} U_2) \\ &= (\phi_2^{(1)}, b_2 - \mathbf{A}_{21} u_1) - (\phi_2^{(1)}, \mathbf{A}_{22} U_2) \\ &= (\phi_2^{(1)}, b_2 - \mathbf{A}_{21} U_1 - \mathbf{A}_{22} U_2) + (\phi_2^{(1)}, \mathbf{A}_{21} e_1) \\ &= (\phi_2^{(1)}, R_2) + (\phi_2^{(1)}, \mathbf{A}_{21} e_1). \end{aligned} \tag{1.70}$$

The first term of the error representation requires only U_2 and $\phi_2^{(1)}$. Since the adjoint system is upper triangular and

$$\phi_2^{(1)} = (\mathbf{A}_{22}^{\mathsf{T}})^{-1} \psi_2^{(1)}$$

is independent of the first component, the calculation of $(\phi_2^{(1)}, R_2)$ remains within the "single physics paradigm", that is we solve the adjoint problem using individual component solves rather than forming and solving a global problem. The second term $(\phi_2^{(1)}, \mathbf{A}_{21} e_1)$ represents the effect of errors in U_1 on the solution

U_2. At first glance this term is uncomputable, but we note that it is a linear functional of e_1 since

$$\left(\phi_2^{(1)}, \mathbf{A}_{21} e_1\right) = \left(\mathbf{A}_{21}^\top \phi_2^{(1)}, e_1\right).$$

We therefore form the adjoint problem for the transfer error,

$$\begin{pmatrix} \mathbf{A}_{11}^\top & \mathbf{A}_{21}^\top \\ \mathbf{0} & \mathbf{A}_{22}^\top \end{pmatrix} \begin{pmatrix} \phi_1^{(2)} \\ \phi_2^{(2)} \end{pmatrix} = \begin{pmatrix} \psi_1^{(2)} \\ 0 \end{pmatrix} = \begin{pmatrix} \mathbf{A}_{21}^\top \phi_2^{(1)} \\ 0 \end{pmatrix}.$$

The upper triangular block structure of $\mathbf{A}^\top$ immediately yields $\phi_2^{(2)} = 0$. As noted earlier, error estimates of u_1 should be independent of u_2. Thus, $\mathbf{A}_{11}^\top \phi_1^{(2)} = \psi_1^{(2)} = \mathbf{A}_{21}^\top \phi_2^{(1)}$, so that once again we can solve for $\phi^{(2)}$ in the "single physics paradigm." Given $\phi^{(2)}$ we obtain the auxiliary error representation

$$\left(\psi^{(2)}, e\right) = \left(\psi_1^{(2)}, e_1\right) = \left(\mathbf{A}_{21}^\top \phi_2^{(1)}, e_1\right) = \left(\mathbf{A}_{11}^\top \phi_1^{(2)}, e_1\right) = \left(\phi_1^{(2)}, R_1\right). \tag{1.71}$$

Combining the first term of (1.70) with (1.71) yields the complete error representation

$$\left(\psi^{(1)}, e\right) = \left(\phi_2^{(1)}, R_2\right) + \left(\phi_1^{(2)}, R_1\right)$$

which is a sum of the inner products of "single physics" residuals and adjoint solutions computed using the "single physics" paradigm.

Example 1.57 We consider an 101×101 system with

$$\begin{aligned} \mathbf{A}_{11} &= I + .2 \times \text{random matrix}, \\ \mathbf{A}_{21} &= \text{random matrix}, \\ \mathbf{A}_{22} &= I + .1 \times \text{random matrix}, \end{aligned}$$

where the coefficients in the random matrices are $U(-1, 1)$. The righthand side is also a random vector with $U(-1, 1)$ coefficients. We solve the linear systems using the Gauss-Seidel iteration with varying numbers of iterations, so that we have control over the accuracy of the solutions. The quantity of interest is the average of the coefficients of u_2.

In the first computation, we solve the first component $\mathbf{A}_{11} U_1 = b_1$ using 20 iterations. The error in the resulting solution is $\approx .031$. When we solve the second component $\mathbf{A}_{22} U_2 = b_2 - \mathbf{A}_{21} U_1$ using 40 or more iterations, we find an error in the quantity of interest of $\approx 3.9 \times 10^{-4}$. We cannot improve the accuracy of the second component regardless of how many iterations we use beyond 40. Solving the adjoint problems, we find

$$\begin{aligned} \left(\psi^{(1)}, e\right) &\approx 3.86 \times 10^{-4}, \\ \left(\phi_2^{(1)}, R_2\right) &\approx 3.1 \times 10^{-8}, \\ \left(\phi_1^{(2)}, R_1\right) &\approx 3.86 \times 10^{-4}. \end{aligned}$$

The error in the quantity of interest is almost entirely due to the error in the solution of the first component.

In the second computation, we solve the first component $\mathbf{A}_{11}U_1 = b_1$ using 200 iterations. The error in the resulting solution is $\approx 10^{-6}$. When we solve the second component $\mathbf{A}_{22}U_2 = b_2 - A_{21}U_1$ using 40 or more iterations, we find an error in the quantity of interest of $\approx 10^{-7}$. This confirms that the error in the solution of the second component in the first computation is dominated by the error in the solution of the first component.

1.6.1.2 *Description of the a posteriori analysis* We seek the error e_2 in a quantity of interest given by a functional of u_2, noting that that a quantity of interest involving u_1 can be computed without solving for u_2. Since we introduce some additional, auxiliary quantities of interest, we denote the primary quantity of interest by $\left(\psi_2^{(1)}, e_2\right)$. We use the adjoint operators

$$\mathcal{A}_1^*(\phi_1, v_1) = (a_1\nabla\phi_1, \nabla v_1) - (\mathrm{div}(b_1\phi_1), v_1) + (c_1\phi_1, v_1)$$
$$\mathcal{A}_2^*(\phi_2, v_2) = (a_2\nabla\phi_2, \nabla v_2) - (\mathrm{div}(b_2\phi_2), v_2) + (c_2\phi_2, v_2).$$

We also use the linearization

$$Lf_2(u_1)(u_1 - U_1) = \int_0^1 \frac{\partial f_2}{\partial u_1}(u_1 s + U_1(1-s))\, ds.$$

Noting that the solution of the first adjoint component is not needed to compute the quantity of interest $(\psi, e) = \left(\psi_2^{(1)}, e_2\right)$, we define the primary adjoint problem to be

$$\mathcal{A}_2^*(\phi_2^{(1)}, v_2) = (\psi_2^{(1)}, v_2), \text{ for all } v_2 \in \mathring{W}_2^1(\Omega).$$

The standard argument yields the error representation,

$$(\psi, e) = \left(\psi_2^{(1)}, e_2\right) = \left(f_2(x, u_1, Du_1), \phi_2^{(1)}\right) - \mathcal{A}_2\left(U_2, \phi_2^{(1)}\right). \tag{1.72}$$

To simplify notation, we denote the weak residual of each solution component by

$$\mathcal{R}_i(U_i, \chi; \nu) = (f_i(\nu), \chi) - \mathcal{A}_i(U_i, \chi), \quad i = 1, 2,$$

so (1.72) becomes

$$(\psi, e) = \mathcal{R}_2(U_2, \phi_2^{(1)}; u_1).$$

At this point, it is not clear that (1.72) is computationally useful since the residual on the righthand side of (1.72) involves the unknown true solution u_1. One consequence is that we cannot immediately use Galerkin orthogonality by inserting a projection of $\phi^{(1)}$ into the representation, since Galerkin orthogonality for U_2 holds for residual $\mathcal{R}_2(U_2, \phi_2^{(1)}; U_1)$ not $\mathcal{R}_2(U_2, \phi_2^{(1)}; u_1)$.

To deal with this, we add and subtract $\left(f_2(x, U_1, DU_1), \phi_2^{(1)}\right)$ in (1.72) and, assuming the same meshes are used for U_1 and U_2, use Galerkin orthogonality to obtain

$$(\psi, e) = \mathcal{R}_2\left(U_2; (I - \Pi_2)\phi_2^{(1)}; U_1\right) + \left(f_2(x, u_1, Du_1) - f_2(x, U_1, DU_1), \phi_2^{(1)}\right), \tag{1.73}$$

where Π_2 is a projection into the finite element space for U_2. The first term on the right of (1.73) is the standard *a posteriori* error expression for the second component while the remaining difference represents the transfer error that arises from using an approximation of u_1 in defining the coefficients in the equation for u_2. The goal now is to estimate this transfer error and its effect on the quantity of interest.

We recognize that the transfer error is a (nominally nonlinear) functional of the error in u_1, defining an auxiliary quantity of interest. We approximate it by a linear functional,

$$\left(f_2(x, u_1, Du_1) - f_2(x, U_1, DU_1), \phi_2^{(1)}\right) \approx \left(Df_2(U_1) \times e_1, \phi_2^{(1)}\right) = \left(e_1, \psi_1^{(2)}\right).$$

We define the corresponding transfer error adjoint problem

$$\mathcal{A}_1^*(\phi_1^{(2)}, v_1) = \left(\psi_1^{(2)}, v_1\right) \text{ for all } v_1 \in \tilde{W}_2^1(\Omega), \tag{1.74}$$

noting that as for the primary problem, we do not have to solve the second component of the full adjoint problem. The transfer error representation formula is given by

$$\left(\psi_1^{(2)}, e_1\right) = \left(f_1, (I - \Pi_1)\phi_1^{(2)}\right) - \mathcal{A}_1(U_1, (I - \Pi_1)\phi_1^{(2)}),$$

where Π_1 is a projection into the finite element space of U_1. We obtain the error representation,

$$\left(\psi, e\right) = \mathcal{R}_2(U_2, (I - \Pi_2)\phi_2^{(1)}; U_1) + \mathcal{R}_1(U_1, (I - \Pi_1)\phi_1^{(2)}). \tag{1.75}$$

In the final step, we account for the error induced by using a multiscale discretization, i.e. different meshes for U_1 and U_2. Example 1.56 shows that this can have a significant effect on overall accuracy.

One issue is that we use

$$f_2(x, \Pi_{1\to 2}U_1, \Pi_{1\to 2}DU_1)$$

in the equations defining the finite element approximation. Correspondingly, we alter the definition of the residual

$$\mathcal{R}_2(U_2, \chi; \nu) = (f_2(\Pi_{1\to 2}\nu), \chi) - \mathcal{A}_2(U_2, \chi).$$

In addition, we use projections to treat any integral involving functions that are defined on both discretizations, i.e. functions of U_i and ϕ_i, $i = 1, 2$. After

decomposing the original estimate to account for all the projections, the new error representation formula for the transfer error becomes

$$\left(Df_2(U_1) \times e_1, \Pi_{2\to 1}\phi_2^{(1)}\right) + \left(Df_2(U_1) \times e_1, (I - \Pi_{2\to 1})\phi_2^{(1)}\right)$$

which is the error contribution arising from the transfer as well as an additional term that is large when the approximation spaces are significantly different.

The data $\psi^{(2)}$ defining the transfer error adjoint is now

$$\left(f_2(u_1) - f_2(U_1), \phi_2^{(1)}\right) \approx \left(Df_2(U_1) \times e_1, \Pi_{2\to 1}\phi_2^{(1)}\right) = \left(\psi_1^{(2)}, e_1\right).$$

The additional term $\left(Df_2(U_1) \times e_1, (I - \Pi_{2\to 1})\phi_2^{(1)}\right)$ is a linear functional, so we define an additional auxiliary quantity of interest

$$\left(\psi_1^{(3)}, e_1\right) = \left(Df_2(U_1) \times e_1, (I - \Pi_{2\to 1})\phi_2^{(1)}\right)$$

and the corresponding adjoint problem

$$\mathcal{A}_1^*(\phi_1^{(3)}, v_1) = \left(\psi_1^{(3)}, v_1\right) \text{ for all } v_1 \in \tilde{W}_2^1(\Omega). \tag{1.76}$$

The final error representation is therefore (Carey *et al.*, 2006)

Theorem 1.58

$$\begin{aligned}(\psi, e) = {} & \mathcal{R}_2(U_2, (I - \Pi_2)\phi_2^{(1)}; U_1) + \mathcal{R}_1(U_1, (I - \Pi_1)(\phi_1^{(2)} + \phi_1^{(3)})) \\ & + \left(\Pi_{1\to 2} f_2(U_1) - f_2(\Pi_{1\to 2}U_1), \phi_2^{(1)}\right) + \left((I - \Pi_{1\to 2})f_2(U_1), \phi_2^{(1)}\right).\end{aligned} \tag{1.77}$$

We emphasize that evaluating the integrals in (1.77) is far from trivial. We have used Monte-Carlo techniques with good results, see (Carey *et al.*, 2006).

Example 1.59 In Example 1.56, we estimate the contributions to the error reported in that computation using the relevant portions of (1.77). To produce the adaptive mesh results shown in Fig. 1.27, we construct the adapted mesh using equidistribution based on a bound derived from the first term in (1.77), i.e. neglecting the terms that estimate the transfer error.

Instead, we consider the system (1.68) for the quantity of interest equal to the average value of U_2. We begin with the same initial coarse meshes as in Fig. 1.27, but add the transfer error expression to the mesh refinement criterion. Adapting the mesh so that the total error in the quantity of interest for U_2 has error estimates less than 10^{-4} yields the meshes shown in Fig. 1.28. We see that the first component solve requires significantly more refinement than the second component.

Example 1.60 This example shows that differences in mesh discretization scale between the two components can contribute significantly to the error. We again solve (1.68) for the quantity of interest equal to $U_2(.15, 15)$. We begin with identical coarse meshes for the two components, but refine only the mesh for U_2. We solve the primary adjoint problem as well as the two auxiliary adjoint problems and show the results in Table 1.3.

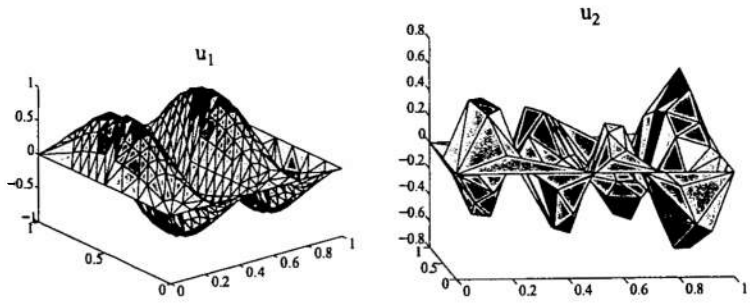


FIG. 1.28. The adapted meshes resulting from the full estimate that accounts for "primary" and "transfer" errors. The transfer error dominates and drives the adaptive refinement.

Primary Error	Transfer Error	Error from Scale Differences
0.000713	0.0905	0.0325

TABLE 1.3. Error contributions when there is a scale difference in the meshes.

1.6.2 *Multiscale decomposition of reaction-diffusion problems*

We follow the presentation in (Estep *et al.*, 2008*a*). In the introduction, we presented operator splitting Alg. 1 for reaction-diffusion problems as a classic example of multiscale operator decomposition. Upon discretizing a reaction-diffusion equation (1.4.4) in space using a standard piecewise linear, continuous finite element method as described in Sec. 1.4.1 we obtain a (high dimensional) initial value problem of the form (1.7). We can then apply the operator splitting algorithm Alg. 1. Finally, we discretize the component problems of the operator splitting method using the either the dG or cG methods on the independent time discretizations $\{t_n\}$, $\{s_{m,n}\}$ described in Fig. 1.4.

For example, if we use the dG methods for both components, then the finite element approximate solutions are sought in a piecewise polynomial spaces for the diffusion and reaction components respectively,

$$\mathcal{V}^{(q_d)} = \left\{ U : U|_{I_n} \in \mathcal{P}^{(q_d)}(I_n), 1 \leq n \leq N \right\},$$
$$\mathcal{V}^{(q_r)}(I_n) = \left\{ U : U|_{I_{m,n}} \in \mathcal{P}^{(q_r)}(I_{m,n}), 1 \leq m \leq M_n \right\},$$

for $n = 1, \cdots, N$, and $I_n = [t_{n-1}, t_n]$ and $I_{m,n} = [s_{m-1,n}, s_{m,n}]$. $\mathcal{P}^{(q_d)}(I_n)$ denotes the space of polynomials in $\mathbb{R}^l$ of degree q_d on I_n. A similar definition holds for $\mathcal{P}^{(q_r)}(I_{m,n})$. We let $U_n^{+,-}$ denote the left- and right-hand limits of U at t_n and $[U]_n = U_n^+ - U_n^-$ the jump value of U at t_n.

Let $\tilde{Y}(t)$ be the piecewise continuous finite element approximation of the operator splitting with

$$\tilde{Y}(t) = \frac{t_n - t}{\Delta t_n}\tilde{Y}_{n-1} + \frac{t - t_{n-1}}{\Delta t_n}\tilde{Y}_n, \quad t_{n-1} \le t \le t_n.$$

The nodal values $\tilde{Y}_n$ are obtained from the following procedure:

Algorithm 3 Multiscale Operator Splitting for Reaction-Diffusion Equations

Set $\tilde{Y}_0 = y_0$
for $n = 1, \cdots, N$ **do**
  Set $Y^r{}_{0,n}^- = \tilde{Y}_{n-1}$
  for $m = 1, \cdots, M_n$ **do**
    Compute $Y^r|_{I_{m,n}} \in \mathcal{P}^{(q_r)}(I_{m,n})$ satisfying

$$\int_{I_{m,n}} \left(\dot{Y}^r, W\right) dt + \left([Y^r]_{m-1,n}, W_{m-1}^+\right) = \int_{I_{m,n}} (F(Y^r), W)\, dt$$
$$\text{for all } W \in \mathcal{P}^{(q_r)}(I_{m,n}) \quad (1.78)$$

  end for
  Set $Y^d{}_{n-1}^- = Y^r{}_{M_n,n}^-$
  Compute $Y^d|_{I_n} \in \mathcal{P}^{(q_d)}(I_n)$ satisfying

$$\int_{I_n} (\dot{Y}^d, V)\, dt + \left([Y^d]_{n-1}, V_{n-1}^+\right) = \int_{I_n} (AY^d, V)\, dt \quad \text{for all } V \in \mathcal{P}^{(q_d)}(I_n) \quad (1.79)$$

  Set $\tilde{y}_n = y^d(t_n^-)$
end for

Adapting standard convergence analysis techniques, we can show that if f is Lipschitz continuous, then for $q_d = 0, 1$ and $q_r = 0, 1$, there exists constants C_1, C_2, C_3 such that,

$$|y_N - \tilde{Y}_N| \le C_1\Delta t + C_2\Delta t^{q_d+1} + C_3\Delta s^{q_r+1}.$$

In Ex. 1.5, we present an example in which multiscale operator decomposition affects the stability, and hence accuracy, of the solution. Such affects can take a myriad of forms.

Example 1.61 In (Estep *et al.*, 2008*a*), we illustrate the instability of operator splitting applied to the Brusselator problem (1.4). We apply a standard first order splitting scheme to a space discretization of the Brusselator model with 500 discrete points with $\alpha = .6$, $\beta = 2$, $k_1 = k_2 = .025$ consisting of the cG(1) method for the diffusion with time step of .2 and dG(0) method for the reaction with time step of .004. On the left of Fig. 1.29, we show a numerical solution that exhibits nonphysical oscillations that developed after some time. On the right, we show plots of the error versus time steps at different times. There is a critical time step above which the instability develops. Moreover, changing the

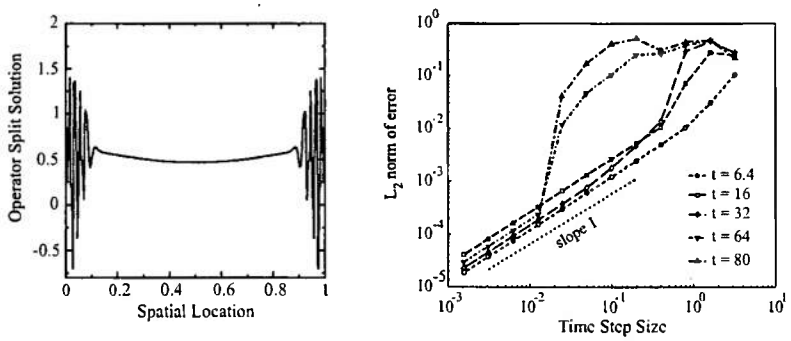


FIG. 1.29. The lefthand plot illustrates typical instability that can arise from multiscale operator splitting applied to Brusselator problem. Solution is shown at time 80. On the right, we show plots of the error in the L_2 norm versus time step size at different times.

space discretization does not improve the accuracy. In fact, using a finer spatial discretization for a constant time step size leads to significantly more error in the long time solution, see (Ropp and Shadid, 2005).

The goal is to derive an accurate *a posteriori* estimate of the error in a specified quantity of interest computed from a multiscale operator splitting approximate solution of (1.7). The estimate must account for the stability effects arising from operator splitting. In the analysis, we distinguish the effects of operator splitting from the effects of numerical discretization of the components. The operator splitting discretization Alg. 3 is a consistent discretization of the formal "analytic" operator splitting Alg. 1 and the numerical error arising in each component can be treated with the standard *a posteriori* analysis discussed previously. Estimating the error arising from the operator splitting itself requires a new approach.

A main technical issue is the definition of a suitable adjoint problem because the standard approach used for nonlinear problems described in Sec. 1.4.3 fails. Indeed, the adjoint operator corresponding to the solution operator for an operator decomposition discretization is typically different than the adjoint operator associated with the true solution operator. Because an adjoint problem carries the global stability information about the quantity of interest, accounting for the differences between adjoint problems associated with the original problem and a numerical discretization are critical for obtaining accurate error estimates.

In the estimate described below, this difference takes the form of "residuals" between certain adjoint operators associated with the fully coupled problem and an analytic operator split version. A practical difficulty with such a result is that solving the adjoint for the fully coupled problem poses the same multiphysics challenges as solving the original forward problem. We therefore develop a new

hybrid *a priori - a posteriori* estimate that combines a computable leading order expression obtained using *a posteriori* arguments with a provably higher order bound obtained using *a priori* convergence result.

1.6.2.1 *A linear algebra example* We illustrate the ideas of the analysis in the context of solving a linear system

$$\mathbf{M}y = b, \tag{1.80}$$

where $\mathbf{M}$ is a $n \times n$ matrix of the form

$$\mathbf{M} = \mathbf{I} - \epsilon(\mathbf{A} + \mathbf{B}),$$

with $\mathbf{I}$ denoting the identity matrix, $\mathbf{A}$ and $\mathbf{B}$ denoting $n \times n$ matrices, and ϵ denotes a small parameter. By standard results, $\mathbf{M}$ is invertible for all sufficiently small ϵ.

The analog of an operator splitting for (1.80) is the problem

$$\mathbf{N}\tilde{y} = b, \tag{1.81}$$

where

$$\mathbf{N} = (\mathbf{I} - \epsilon\mathbf{A})(\mathbf{I} - \epsilon\mathbf{B})$$

is also invertible for ϵ small. This follows from the observation

$$\mathbf{M}^{-1} = \big(\mathbf{I} - \epsilon(\mathbf{A} + \mathbf{B})\big)^{-1}, \quad \mathbf{N}^{-1} = \big(\mathbf{I} - \epsilon\mathbf{B}\big)^{-1}\big(\mathbf{I} - \epsilon\mathbf{A}\big)^{-1},$$

so that inverting $\mathbf{N}$ involves inverting operators involving only $\mathbf{A}$ and $\mathbf{B}$ individually. Using the Neumann series to represent the inverse operators leads to the estimate,

$$\begin{aligned}\mathbf{M}^{-1} - \mathbf{N}^{-1} &= \mathbf{I} + \epsilon(\mathbf{A} + \mathbf{B}) + \epsilon^2(\mathbf{A} + \mathbf{B})^2 + \epsilon^3(\mathbf{A} + \mathbf{B})^3 + \cdots \\ &\quad - \big(\mathbf{I} + \epsilon\mathbf{B} + \epsilon^2\mathbf{B}^2 + \epsilon^3\mathbf{B}^3 + \cdots\big) \times \big(\mathbf{I} + \epsilon\mathbf{A} + \epsilon^2\mathbf{A}^2 + \epsilon^3\mathbf{A}^3 + \cdots\big) \\ &= \epsilon^2\mathbf{B}\mathbf{A} + \mathbf{O}(\epsilon^3).\end{aligned} \tag{1.82}$$

We consider the problem of solving (1.80) to compute the quantity of interest (y, ψ). We have the associated adjoint problems

$$\mathbf{M}^*\phi = \psi, \quad \mathbf{N}^*\tilde{\phi} = \psi.$$

If $\tilde{Y} \approx \tilde{y}$ is a numerical solution, the standard *a posteriori* analysis in Ex. 1.41 gives

$$(\tilde{e}, \psi) = (\tilde{Y} - \tilde{y}, \psi) = (\tilde{\phi}, \tilde{R}), \quad \tilde{R} = \mathbf{N}\tilde{Y} - b.$$

Since we assume that problems involving N are solvable, this is a computable estimate. However, the error we wish to estimate is $(\tilde{Y} - y, \psi)$. We write this as

$$(\tilde{Y} - y, \psi) = (\tilde{Y} - \tilde{y}, \psi) + (\tilde{y} - y, \psi) = (\tilde{\phi}, \tilde{R}) + (\tilde{y} - y, \psi).$$

The second term on the right is the error arising from the operator splitting. Since these problems are linear, we can use the Greens function formulas

$$(y, \psi) = (\phi, b), \quad (\tilde{y}, \psi) = (\tilde{\phi}, b),$$

to conclude

$$(\tilde{Y} - y, \psi) = (\tilde{\phi}, \tilde{R}) + (\tilde{\phi} - \phi, b). \tag{1.83}$$

Example 1.62 We let $\mathbf{A}$ and $\mathbf{B}$ be random 500×500 matrices, where the coefficients in the random matrices are $U(-1, 1)$, normalized in the 2-norm and $\epsilon = .01$. We set y to be a random vector, with $U(-1, 1)$ coefficients, and set $b = My$, which insures that y is a solution within machine precision. Finally, we set $\psi = (1, 0, \cdots)^\top$. We compute $\tilde{Y}$ using Gaussian elimination. We find

$$(\tilde{Y} - y, \psi) \approx 5.376 \times 10^{-5}$$
$$(\tilde{\phi}, \tilde{R}) \approx 1.221 \times 10^{-15}$$
$$(\tilde{\phi} - \phi, b) \approx 5.376 \times 10^{-5}.$$

This means that nearly all of the error is captured by the effect of operator splitting on the adjoint solution.

As noted above, (1.83) is problematic because it requires the solution of the "true" adjoint problem, which is unavailable in the operator splitting paradigm.

1.6.2.2 *Description of the hybrid a posteriori-a priori error analysis* We now describe an error estimate for a multiscale operator decomposition solution of (1.7) that is composed of a leading expression is *a posteriori* and an *a priori* expression that is provably higher order. See (Estep *et al.*, 2008*a*) for the full analysis.

We begin with the decomposition

$$\tilde{Y} - y = (\tilde{Y} - \tilde{y}) + (\tilde{y} - y), \tag{1.84}$$

where y solves (1.7), $\tilde{y}$ is computed via the abstract operator splitting Alg. 1, and $\tilde{Y}$ is computed via the numerical operator splitting Alg. 3.

The first expression on the right of (1.84) is the error of $\tilde{Y}$ as a solution of the operator split problem. Note that $\tilde{Y}$ is a consistent numerical solution for the analytic operator split problem and the expression for its error can be estimated using the standard *a posteriori* error analysis. We let ϑ^d define the adjoint solution associated with the diffusion component (1.79) satisfying

$$\begin{cases} -\dot{\vartheta}^d = A^\top \vartheta^d(t), & t_n > t \geq t_{n-1}, \\ \vartheta^d(t_n^-) = \psi_n. \end{cases}$$

Furthermore, we let ϑ^r define the adjoint solution associated with the reaction component (1.78) satisfying

$$\begin{cases} -\dot{\vartheta}^r = (\hat{F}'(y^r, Y^r))^{\mathsf{T}} \vartheta^r(t), & s_{m,n} > t \geq s_{m-1,n}; \\ \vartheta^r(s_{m,n}) = \psi^r_{m,n}; \end{cases}$$

for $m = M_n, \cdots, 1$, with $\psi^r{}_{M_n,n} = \vartheta^{d+}_{n-1}$ and $\psi^r{}_{m,n} = \vartheta^r_{m,n}$ for $m < M_n$. Thus ϑ^r is continuous across the internal reaction time nodes $s_{m,n}$, $m = 1, \cdots, M_n - 1$. Here,

$$\hat{F}'(y^r, Y^r) = \int_0^1 F'(sy^r + (1-s)Y^r)\, ds.$$

The second expression on the right of (1.84) is an abstract error of operator splitting. Following the analysis for the linear algebra example, we use analogs of the classic representation formula involving the Greens function of a linear elliptic problem to construct an estimate. The nonlinearity complicates the analysis however because we have to use linearization to define unique adjoint problems, which raises the issue of choosing a trajectory around which to linearize. We cannot use the standard approach of linearizing the error representation described in Sec. 1.4.3 because of the operator splitting. Instead, we assume that both the original problem and the operator split version have a common solution and we linearize each problem in a neighborhood of this common solution. For example, we assume that $y = 0$ is a steady state solution of both problems, which can be achieved by assuming that

$$\text{Homogeneity Assumption:} \quad F(0) = 0,$$

and we linearize in a region around 0. In terms of applications to reaction-diffusion problems, there are mathematical reasons for making the homogeneity assumption and it is satisfied in a great many applications. However, we can modify the analysis to allow for linearization around any known common solution, see (Estep *et al.*, 2008*a*).

To motivate this definition, we derive an abstract Greens function representation. On time interval (t_{n-1}, t_n), we consider the linearized problem,

$$\begin{cases} \dot{y} = A\, y(t) + \overline{F'(y)}\, y(t), & t_{n-1} < t \leq t_n; \\ y(t_{n-1}) = y_{n-1}; \end{cases}$$

where

$$\overline{F'(y)} = \int_0^1 F'(sy)\, ds.$$

We note that $\overline{F'(y)}y = F(y)$ because $F(0) = 0$. The generalized Greens function φ satisfies the adjoint problem

$$\begin{cases} -\dot{\varphi} = A^{\mathsf{T}} \varphi(t) + \overline{F'(y)}^{\mathsf{T}} \varphi(t), & t_n > t \geq t_{n-1}, \\ \varphi(t_n) = \psi_n; \end{cases} \tag{1.85}$$

where ψ_n determines the quantity of interest $(y(t_n), \psi_n)$, and A^{T} and $\overline{F'(y)}^{\mathsf{T}}$ denote the transpose of A and $\overline{F'(y)}$, respectively. We choose $\psi_n = \varphi(t_n^+)$, which

couples the local adjoint problems (1.85) to form a global adjoint problem. This definition yields a simple representation of the solution value over one time step

$$(y_n, \psi_n) = (y_{n-1}, \varphi_{n-1}),\ n = 1, 2, \cdots, N \implies (y_N, \psi_N) = (y_0, \varphi_N). \quad (1.86)$$

We use analogs for (1.86) for solutions of each component in the operator splitting discretization. For $n = 1, \cdots, N$, we define the three adjoint problems. The diffusion problem is simpler because it is linear,

$$\begin{cases} -\dot{\varphi}^d = A^\top \varphi^d(t), & t_n > t \geq t_{n-1}, \\ \varphi^d(t_n^-) = \psi_n^d. \end{cases} \quad (1.87)$$

It is convenient to let Φ_n^d denote the solution operator, so $\varphi^d(t_{n-1}) = \Phi_n^d \psi_n^d$. We require two adjoint problems to treat the reaction component. The difference between the problems is the function around which they linearized,

$$\begin{cases} -\dot{\varphi}_1^r = \overline{F'(\tilde{Y})}^\top \varphi_1^r(t), & t_n > t \geq t_{n-1}, \\ \varphi_1^r(t_n^-) = \psi_n^r. \end{cases} \quad (1.88)$$

$$\begin{cases} -\dot{\varphi}_2^r = \overline{F'(Y^r)}^\top \varphi_2^r(t), & t_n > t \geq t_{n-1}, \\ \varphi_2^r(t_n^-) = \psi_n^r. \end{cases} \quad (1.89)$$

If $\Phi_n^r(z)$ denotes the solution operator for the problem linearized around a function z, then we have $\varphi_1^r(t_{n-1}) = \Phi_n^r(\tilde{Y})\psi_n^r$ and $\varphi_2^r(t_{n-1}) = \Phi_n^r(Y^r)\psi_n^r$. We can now prove (Estep *et al.*, 2008*a*).

Theorem 1.63 *A hybrid* a posteriori - a priori *error estimate for the multiscale operator splitting dG finite element method is*

$$\begin{aligned} (\tilde{Y}_N - y_N, \psi_N) = &\sum_{n=1}^{N} \sum_{m=1}^{M_n} \left(\int_{I_{m,n}} (\dot{Y}^r - F(Y^r), \vartheta^r - \Pi\vartheta^r)\, dt \right. \\ &\left. + ([Y^r]_{m-1,n}, \vartheta^{r+}_{m-1,n} - \Pi\vartheta^{r+}_{m-1,n}) \right) \\ &+ \sum_{n=1}^{N} \left(\int_{I_n} (\dot{Y}^d - AY^d, \vartheta^d - \Pi\vartheta^d)\, dt \right. \\ &\left. + ([Y^d]_{n-1}, \vartheta^{d+}_{n-1} - \Pi\vartheta^{d+}_{n-1}) \right) \\ &+ \sum_{n=1}^{N} (\tilde{Y}_{n-1}, (E_1 + E_2)\psi_n) + \mathbf{O}(\Delta t^{q_d+2}) + \mathbf{O}(\Delta t\, \Delta s^{q_r+1}), \end{aligned}$$

where

$$E_1 = \frac{1}{2}\Delta t_n \left(A^\top \mathcal{F}(\tilde{Y}) - \mathcal{F}(\tilde{Y}) A^\top \right),\ \mathcal{F}(\tilde{Y}) = \int_{I_n} \overline{F'(\tilde{Y})}\, dt,$$

$$E_2 = \left(\Phi_n^r(\tilde{Y}) - \Phi_n^r(Y^r) \right) \Phi_n^d.$$

The first expression on the right is the error introduced by the numerical solution of the reaction component. Likewise, the second expression on the right is the error introduced by the numerical solution of the diffusion component. The third expression on the right measures the effects of operator splitting. The expression E_1 is a leading order estimate for the effects of operator splitting while E_2 accounts for issues arising from the differences in linearizing around the global computed solution as opposed to the solution of the reaction component, which affects the formulation of the adjoint problems. Both of these quantities are scaled by the solution itself, so that these effects become negligible when the solution approaches zero. Finally, the remaining terms represent bounds on terms that are not computable but are higher order. In practice, we neglect those terms when computing an estimate.

Using the estimate requires the solution of five adjoint problems. But we avoid the need to solve an adjoint problem corresponding to linearization around the true solution by deriving the hybrid estimate.

1.6.2.3 *Numerical examples* We describe some examples in (Estep *et al.*, 2008*a*).

Example 1.64 The first example is partial differential equation version of Ex. 1.5,

$$\begin{cases} \frac{\partial u}{\partial t} - 0.05\,\frac{\partial^2 u}{\partial x^2} = u^2, & x \in (0,1), t > 0, \\ u(0,t) = u(1,t) = 0, & t > 0, \\ u(x,0) = 4x(1-x), & x \in (0,1). \end{cases}$$

The solution of the reaction component exhibits finite time blow up when undamped by the diffusion component. This is perhaps the most extreme form of instability. For this computation, we use 20 spatial finite elements. Table 1.4 shows the ratio of the error to the estimate computed at the final time $T = 1$. In this computation, we keep the reaction time step constant and vary the diffusion time step and number of reaction time steps. We see that the estimate is very accurate for a range of time steps.

TABLE 1.4. Operator splitting error estimate for the blow up problem at $T = 1$, reaction time step $= 10^{-3}$

Δt	M	Exact Err (%)	Error/Estimate
10^{-1}	100	11.07	1.0286
10^{-2}	10	1.35	1.0067
10^{-3}	1	0.45	1.0020

Example 1.65 We next consider the Brusselator problem (1.4) with $\alpha = 2$, $\beta = 5.45$, $k_1 = 0.008$, $k_2 = 0.004$ and initial conditions $u_1(x,0) = \alpha + 0.1\sin(\pi x)$ and $u_2(x,0) = \beta/\alpha + 0.1\sin(\pi x)$, which yields an oscillatory solution. In this case, the reaction is very mildly unstable, with at most polynomial rate accumulation of perturbations as time passes. We use a 32 node spatial finite element

discretization, resulting in an differential equation system with dimension 62. We note that in original form, the reaction terms do not satisfy the requirement $F(0) = 0$ so we linearize around the steady state solution c with with $c_i = \alpha$ for $i = 1, \cdots, N_e - 1$ and $c_i = \beta/\alpha$ for $i = N_e, \cdots, 2N_e - 2$, so that $F(c) = 0$.

Fig. 1.30 compares the errors computed using $\Delta t = 0.01$ and $M = 10$ reaction time steps to the hybrid *a posteriori* error estimates. We show results for $[0, 2]$, when the solution is still in a transient stage, and at $T = 40$ when the solution has become periodic. All the results show that the exact and estimated errors are in remarkable agreement.

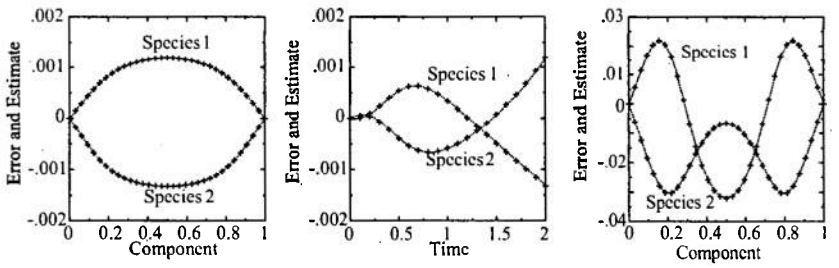


FIG. 1.30. Brusselator results. Left: Comparison of errors against the spatial location at $T = 2$. Middle: Time history of errors at the midpoint location on $[0, 2]$. Right: Comparison of errors against the spatial location at $T = 40$. The dotted line is the exact error and the (+) is the estimated error

1.6.3 *Multiscale decomposition of a fluid-solid conjugate heat transfer problem*

Following (Estep *et al.*, 2008*b*), we next consider the multiscale decomposition solution of the heat transfer problem described in Example 1.3. The weak formulation of (1.5) consists of computing $u \in V_F$, $p \in L_0^2(\Omega_F)$, $T_F \in W_F$ and $T_S \in W_S$ such that

$$\begin{cases} a_1(u, v) + c_1(u, u, v) + b(v, p) + d(T_F, v) = (f, v), \\ b(u, q) = 0, \\ a_2(T_F, w_F) + c_2(u, T_F, w_F) + a_3(T_S, w_S) = (Q_F, w_F) + (Q_S, w_S), \end{cases} \tag{1.90}$$

for all $v \in V_{F,0}$, $q \in L_0^2(\Omega_F)$, $w_F \in W_{F,0}$ and $w_S \in W_{S,0}$, where

$$\begin{aligned} &a_1(u, v) = \textstyle\int_{\Omega_F} \mu(\nabla u : \nabla v)\, dx, && a_2(T_F, w_F) = \textstyle\int_{\Omega_F} k_F(\nabla T_F \cdot \nabla w_F)\, dx, \\ &a_3(T_S, w_S) = \textstyle\int_{\Omega_S} k_S(\nabla T_S \cdot \nabla w_S)\, dx, && b(v, q) = -\textstyle\int_{\Omega_F} (\nabla \cdot v) q\, dx, \\ &c_1(u, v, z) = \textstyle\int_{\Omega_F} \rho_0\, (u \cdot \nabla)\, v \cdot z\, dx, && c_2(u, T, w) = \textstyle\int_{\Omega_F} \rho_0 c_p (u \cdot \nabla T)\, w\, dx, \\ &d(T, v) = \textstyle\int_{\Omega_F} \rho_0 \beta T g \cdot v\, dx, && f = \rho_0\, (1 + \beta T_0)\, g, \end{aligned}$$

and

$$V_F = \{v \in H^1(\Omega_F) \,|\, v = g_{u,D} \text{ on } \Gamma_{u,D}\}, \qquad V_{F,0} = \{v \in V_F \,|\, v = 0 \text{ on } \Gamma_{u,D}\},$$
$$W_F = \{w \in H^1(\Omega_F) \,|\, w = g_{T_F,D} \text{ on } \Gamma_{T_F,D}\}, \quad W_{F,0} = \{w \in W_F \,|\, w = 0 \text{ on } \Gamma_{T_F,D}\},$$
$$W_S = \{w \in H^1(\Omega_S) \,|\, w = g_{T_S,D} \text{ on } \Gamma_{T_S,D}\}, \quad W_{S,0} = \{w \in W_S \,|\, w = 0 \text{ on } \Gamma_{T_S,D}\}.$$

To discretize, we construct independent locally-quasi-uniform triangulations $T_{F,h}$ and $T_{S,h}$ of Ω_F and Ω_S respectively. We use the piecewise polynomial spaces

$$V_F^h = \{v \in V_F \mid v \text{ continuous on } \Omega_F;\ v_i \in P^2(K) \text{ for all } K \in \tau_{F,h}\},$$
$$Z^h = \{z \in Z \mid z \text{ continuous on } \Omega_F;\ z \in P^1(K) \text{ for all } K \in \tau_{F,h}\},$$
$$W_F^h = \{w \in W_F \mid w \text{ continuous on } \Omega_F,\ w \in P^2(K) \text{ for all } K \in \tau_{F,h}\},$$
$$W_S^h = \{w \in W_S \mid w \text{ continuous on } \Omega_S;\ w \in P^2(K) \text{ for all } K \in \tau_{S,h}\},$$

and the associated subspaces

$$V_{F,0}^h = \{v \in V^h \mid v = 0 \text{ on } \Gamma_{u,D}\},$$
$$W_{F,0}^h = \{w \in W_F^h \mid w = 0 \text{ on } \Gamma_{T_F,D}\},$$
$$W_{S,0}^h = \{w \in W_S^h \mid w = 0 \text{ on } \Gamma_{T_S,D} \text{ and } w = 0 \text{ on } \Gamma_I\},$$

where $P^q(K)$ denotes the space of polynomials of degree q on an element K. Note that $W_{S,0}^h$ is different than $W_{F,0}^h$ in an important way since $\Gamma_{T_S,D}$ does not include Γ_I. We let π_V, π_{W_F}, π_{W_S}, and π_Z be projections into V_F^h, W_F^h, W_S^h and Z^h respectively. We also use π_{W_F} and π_{W_S} to denote projections into W_F^h and W_S^h respectively along the interface Γ_I.

Algorithm 4 Multiscale Decomposition Method for Conjugate Heat Transfer

k = 0
while $(\|T_S^{\{k\}} - \pi_S T_F^{\{k\}}\|_{\Gamma_I} > TOL)$ **do**
k = k+1
Compute $T_{S,h}^{\{k\}} \in W_S^h$ such that $T_{S,h}^{\{k\}} = \pi_{W_S} T_{F,h}^{\{k-1\}}$ along the interface Γ_I and

$$a_3(T_{S,h}^{\{k\}}, w) = (Q_S, w), \quad \forall w \in W_{S,0}^h, \tag{1.91}$$

Compute $u_h^{\{k\}} \in V_F^h$, $p_h^{\{k\}} \in Z^h$ and $T_{F,h}^{\{k\}} \in W_F^h$ such that

$$\begin{cases} a_1(u_h^{\{k\}}, v) + c_1(u_h^{\{k\}}, u_h^{\{k\}}, v) + b(v, p_h^{\{k\}}) + d(T_{F,h}^{\{k\}}, v) = (f, v), \ \forall v \in V_{F,0}^h, \\ b(u_h^{\{k\}}, q) = 0, \ \forall q \in Z^h, \\ a_2(T_{F,h}^{\{k\}}, w) + c_2(u_h^{\{k\}}, T_{F,h}^{\{k\}}, w) = (Q_F, w) - (k_S(n \cdot \nabla T_{S,h}^{\{k\}}), w)_{\Gamma_I}, \\ \qquad\qquad \forall w \in W_{F,0}^h. \end{cases} \tag{1.92}$$

end while

To compute a stable solution of the fluid equations, we choose V_F^h and Z^h to be the Taylor-Hood finite element pair satisfying the discrete inf-sup condition

$$\inf_{q\in Z^h}\sup_{v\in V_F^h}\frac{b(v,q)}{\|v\|_1\cdot\|q\|_0}\geq\beta>0. \tag{1.93}$$

We also note that the convergence of the iteration defined by this Algorithm depends on the values of k_S and k_F along the interface and the geometry of each region. Often, a "relaxation" is used to help improve convergence properties. We choose $\alpha\in[0,1)$ and impose the relaxed Dirichlet interface values

$$T_F^{\{k\}}=\alpha T_F^{\{k-1\}}+(1-\alpha)\pi_F T_S^{\{k-1\}}.$$

This affects the analysis, but we do not discuss that here, see (Estep *et al.*, 2008*a*; Estep *et al.*, 2008*b*).

1.6.3.1 *Description of an a posteriori error analysis* We define the adjoint using the standard linearization approach. We define the errors

$$e_u=u-u_h^{\{k\}},e_p=p-p_h^{\{k\}},e_{T_F}=T_F-T_{F,h}^{\{k\}}\text{ and }e_{T_S}=T_S-T_{S,h}^{\{k\}}.$$

The adjoint problem for the quantity of interest

$$(\psi,e)=(\psi_u,e_u)+(\psi_p,e_p)+(\psi_{T_F},e_{T_F})+(\psi_{T_S},e_{T_S})$$

for the coupled problem (1.5) is

$$\begin{cases}-\mu\Delta\phi+\overline{c}_1^*(\phi)+\nabla z+\overline{c}_{2u}^*(\theta_F)=\psi_u, & x\in\Omega_F,\\ -\nabla\cdot\phi=\psi_p, & x\in\Omega_F,\\ -k_F\Delta\theta_F+\overline{c}_{2T}^*(\theta_F)+\rho_0\beta(g\cdot\phi)=\psi_{T_F}, & x\in\Omega_F,\\ \begin{cases}\theta_F=\theta_S,\\ k_F(n\cdot\nabla\theta_F)=k_S(n\cdot\nabla\theta_S),\end{cases} & x\in\Gamma_I,\\ -k_S\Delta\theta_S=\psi_{T_S}, & x\in\Omega_S,\end{cases} \tag{1.94}$$

with adjoint boundary conditions

$$\begin{cases}\phi=0, & x\in\Gamma_{u,D},\\ \mu\dfrac{\partial\phi}{\partial n}=0, & x\in\Gamma_{u,N},\\ \theta_F=0, & x\in\Gamma_{T_F,D},\\ k_F(n\cdot\nabla\theta_F)=0, & x\in\Gamma_{T_F,N},\\ \theta_S=0, & x\in\Gamma_{T_S,D},\\ k_S(n\cdot\nabla\theta_S)=0, & x\in\Gamma_{T_S,N}.\end{cases} \tag{1.95}$$

Here, we have used the linearizations

$$\begin{aligned}\overline{c}_1^*(\phi)&=\frac{1}{2}\rho_0\nabla(u+u_h)\cdot\phi-\frac{1}{2}\rho_0(u+u_h)\cdot\nabla\phi-\frac{1}{2}\rho_0(\nabla\cdot(u+u_h))\phi,\\ \overline{c}_{2u}^*(\theta)&=\frac{1}{2}\rho_0c_p\nabla(T+T_h)\theta,\\ \overline{c}_{2T}^*(\theta)&=-\frac{1}{2}\rho_0c_p(u+u_h)\cdot\nabla\theta-\frac{1}{2}\rho_0c_p(\nabla\cdot(u+u_h))\theta.\end{aligned}$$

We solve (1.94) numerically using an iterative operator decomposition approach as for the forward problem. These iterations are completely independent of the forward iterations. In (Estep *et al.*, 2008*a*; Estep *et al.*, 2008*b*), we derive estimates that only require adjoint solutions of the two component problems.

To write out the *a posteriori* error representation, we introduce an additional projection $\pi^0_{W_S} : H^2 \to W^h_{S,0}$ defined such that for any node x_i

$$\pi^0_{W_S}\theta_S(x_i) = \begin{cases} \pi_{W_S}\theta_S(x_i), & x_i \notin \Gamma_I, \\ 0, & x_i \in \Gamma_I, \end{cases}$$

along with $\pi_\partial \theta_S = \pi_{W_S}\theta_S - \pi^0_{W_S}\theta_S$. The role of these projections is made clear in the context of improving accuracy, see (Estep *et al.*, 2008*b*) and remarks below.

We can now prove (Estep *et al.*, 2008*b*),

Theorem 1.66 *The errors satisfy*

$$\begin{aligned}(\psi, e) = {} & (f, \phi - \pi_V\phi) - a_1(u_h^{\{k\}}, \phi - \pi_V\phi_1) - c_1(u_h^{\{k\}}, u_h^{\{k\}}, \phi - \pi_V\phi) \\ & - b(\phi - \pi_V\phi, p_h) - d(T_{F,h}^{\{k\}}, \phi - \pi_V\phi) - b(u_h^{\{k\}}, z - \pi_Z z) \qquad (1.96) \\ & + (Q_F, \theta_F - \pi_{W_F}\theta_F) - a_2(T_{F,h}^{\{k\}}, \theta_F - \pi_{W_F}\theta_F) \\ & - c_2(u_h^{\{k\}}, T_{F,h}^{\{k\}}, \theta_F - \pi_{W_F}\theta_F) + (Q_S, \theta_S - \pi_{W_S}\theta_S) \\ & - a_3(T_{S,h}^{\{k\}}, \theta_S - \pi_{W_S}\theta_S) \qquad (1.97) \\ & + \left(T_{S,h}^{\{k\}} - \pi_S T_{F,h}^{\{k\}}, k_S(n \cdot \nabla\theta_S)\right)_{\Gamma_I} + \left(\pi_S T_{F,h}^{\{k\}} - T_{F,h}^{\{k\}}, k_S(n \cdot \nabla\theta_S)\right)_{\Gamma_I} \\ & \qquad (1.98) \\ & + \left(k_S(n \cdot \nabla T_{S,h}^{\{k\}}), \pi_{W_F}\theta_F\right)_{\Gamma_I} + \left(Q_S, \pi_{W_S}\theta_S\right) - a_3\left(T_{S,h}^{\{k\}}, \pi_{W_S}\theta_S\right). \qquad (1.99)\end{aligned}$$

The contributions to the error are

- Equations (1.96)-(1.97) represents the contribution of the discretization error arising from each component solve.
- Equation (1.98) represents the contribution from the iteration.
- The first term in (1:99) represents contribution of the transfer error while the remaining terms represent the contribution arising from projections between two different discretizations.

Example 1.67 We consider an example from (Estep *et al.*, 2008*b*). For the flow past a cylinder shown in Figure 1.2, we solve the steady non-dimensionalized Boussinesq equations in the fluid domain and the non-dimensional heat equation in the solid domain. To simulate the flow of water past a cylinder made from stainless steel, we set the dimensionless constants $Pr = 6.6$ and $k_R = 30$, and choose the inflow velocity and the temperature gradient so that,

$$Re = 75, \quad Pe = 495, \quad Fr = 0.001, \quad Ra = 50.$$

The temperature gradient is imposed by setting different temperatures along the top and bottom boundaries, with a linear temperature gradient on the inflow boundary, and an adiabatic condition on the outflow boundary.

We show results for two quantities of interest. The first is the temperature in a small region in the wake, located approximately one channel width downstream of the center of the cylinder and 1/4 of a channel width below the upper wall. The second is temperature at a small region in the center of the cylinder. In each case, we derive an *a posteriori* bound by the usual methods, and base adaptivity on an element tolerance of 1×10^{-8}.

We show the final adaptive meshes for the flow in Fig. 1.31 and for the solid in Fig. 1.32. For the first quantity of interest, the flow mesh is most refined near the region of interest and upstream of the region of interest, locating more elements between the cylinder and the top wall than the cylinder and the bottom wall since the flow advecting heat to the region of interest passes above rather than below the cylinder. The solution downstream of the region of interest can be computed with less accuracy as is recognized by the coarser mesh. For the solid, the mesh is highly refined along the top in order to increase the accuracy of the normal derivative that is computed in the solid and used as a boundary condition in the fluid computation. Evidently, the normal derivatives elsewhere on the interface have less of an influence on the first quantity of interest. For the

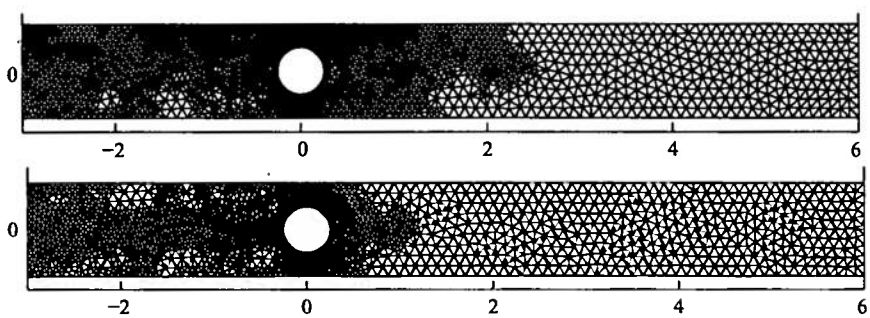


FIG. 1.31. Upper: Final adaptive mesh in the fluid when the quantity of interest is the temperature in a small region in the wake above the cylinder. Lower: Final adaptive mesh in the fluid when the quantity of interest is the temperature in a small region in the center of the solid.

second quantity of interest, the mesh is highly refined upstream of the cylinder. We note that the refinement downstream of the cylinder corresponds closely to the recirculation region, and the mesh refinement is slightly asymmetric about the midplane of the channel due to the asymmetric initial mesh. The mesh in the solid is refined uniformly near the boundary, reflecting the fact that the error in the finite element flux makes a significant contribution to the error in the

quantity of interest.

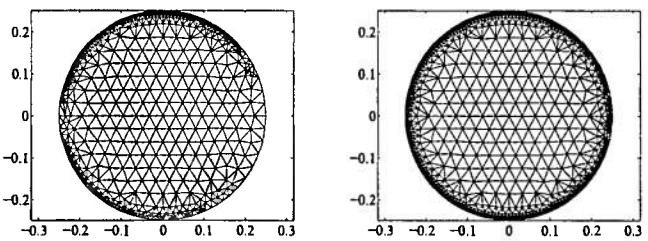


FIG. 1.32. Left: Final adaptive mesh in the solid when the quantity of interest is the temperature in a small region in the wake above the cylinder. Right: Final adaptive mesh in the solid when the quantity of interest is the temperature in a small region in the center of the solid.

1.6.3.2 *Loss of order and flux correction* The meshes shown in Fig. 1.31 and Fig. 1.32 are highly refined near the interface. This reflects the fact that there is significant error in the numerical flux passed between the components. It turns out that this pollutes the entire computation, so that overall the method loses an entire order of accuracy.

Example 1.68 We apply Alg. 4 to the steady flow of a Newtonian fluid in a two-dimensional channel connected along one boundary to a solid which is heated from below as shown in Fig. 1.33.

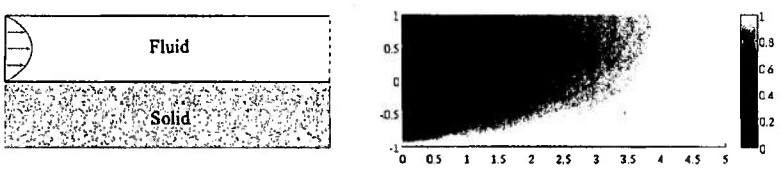


FIG. 1.33. Left: Computational domain for motivational example. Right: Temperature fields within the fluid and the solid.

The Reynolds number (based on the channel width and the flux averaged inlet velocity) is $Re = 2.5$ and the thermal conductivities are $k_F = 0.9$ and $k_S = 1 + 0.5\sin(2\pi x)\sin(2\pi y)$, which are chosen so that the solution is smooth, but nontrivial. The temperature fields are displayed in Fig. 1.33.

We solve the problem iteratively and, to approximate the error, we compute a reference solution with a higher order method on the same mesh. In Fig. 1.34, we compare the L^2 errors in the temperature fields over $\Omega_S \cup \Omega_F$ on a series of

meshes that align along the interface Γ_I.

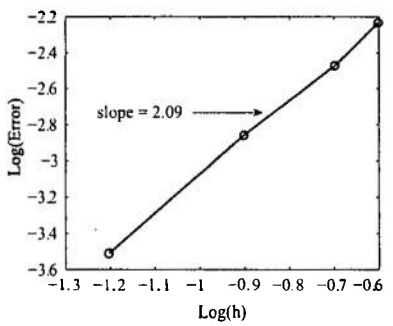


FIG. 1.34. Comparison of the mesh size, h, versus the L^2 error in the temperature field when the finite element flux is passed.

We see that the solution converges at a second order rate, rather than the optimal third order rate. This loss of order is a consequence of the operator decomposition as the computed boundary flux obtained from the finite element solution is one order less accurate than the solution itself and this error pollutes the rest of the computation.

One way to compensate for the loss of order is by refining the mesh locally near the interface. Another way is to compute the particular information, in this case the flux on the interface, more accurately. It turns out that we can adapt a post-processing technique called flux correction developed originally by Wheeler (Wheeler, 1974) and Carey (G.F. Carey, 1985; Carey, 1982) to recover boundary flux values with increased accuracy.

We denote the set of elements in $\tau_{S,h}$ that intersect the interface boundary by

$$\tau_{S,h}^{\Gamma_I} = \{K \in \tau_{S,h} \mid \overline{K} \cap \Gamma_I \neq \emptyset\},$$

and consider the corresponding finite element space

$$\Sigma_h = \{v \in P^2(K) \text{ with } K \in \tau_{S,h}^{\Gamma_I},\ v(\eta_i) = 0 \text{ if } \eta_i \notin \Gamma_I\},$$

where $\{\eta_i\}$ denotes the nodes of element K. The degrees of freedom correspond to the nodes on the boundary. We compute $\sigma^{\{k\}} \in \Sigma_h$ satisfying

$$-\left(\sigma^{\{k\}}, v\right)_{\Gamma_I} = (Q_S, v) - a_3(T_{S,h}^{\{k\}}, v), \quad \text{for all } v \in \Sigma_h,$$

where $T_{S,h}^{\{k\}}$ solves (1.91). Since the dimension of the problem scales with the number of nodes on a boundary, it is relatively inexpensive to solve.

The modified Algorithm is given in Alg. 5.

Algorithm 5 Multiscale Decomposition Method for Conjugate Heat Transfer with Flux Correction

k = 0
while $(\|T_S^{\{k\}} - \pi_S T_F^{\{k\}}\|_{\Gamma_I} > TOL)$ **do**
k = k+1
Compute $T_{S,h}^{\{k\}} \in W_S^h$ such that $T_{S,h}^{\{k\}} = \pi_{W_S} T_{F,h}^{\{k-1\}}$ along the interface Γ_I and

$$a_3(T_{S,h}^{\{k\}}, w) = (Q_S, w), \quad \forall w \in W_{S,0}^h, \tag{1.100}$$

Compute $\sigma^{\{k\}} \in \Sigma_h$ solving

$$-\left(\sigma^{\{k\}}, v\right)_{\Gamma_I} = (Q_S, v) - a_3(T_{S,h}^{\{k\}}, v), \ \forall v \in \Sigma_h, \tag{1.101}$$

Compute $u_h^{\{k\}} \in V_F^h$, $p_h^{\{k\}} \in Z^h$ and $T_{F,h}^{\{k\}} \in W_F^h$ such that

$$\begin{cases} a_1(u_h^{\{k\}}, v) + c_1(u_h^{\{k\}}; u_h^{\{k\}}, v) + b(v, p_h^{\{k\}}) + d(T_{F,h}^{\{k\}}, v) = (f, v), \ \forall v \in V_{F,0}^h, \\ b(u_h^{\{k\}}, q) = 0, \ \forall q \in Z^h, \\ a_2(T_{F,h}^{\{k\}}, w) + c_2(u_h^{\{k\}}, T_{F,h}^{\{k\}}, w) = (Q_F, w) - \left(\sigma^{\{k\}}, w\right)_{\Gamma_I}, \ \forall w \in W_{F,0}^h. \end{cases} \tag{1.102}$$

end while

It turns out that using the recovered boundary flux leads to a cancelation of the "transfer error" term in the error representation formula, which is the source of the loss of order. The new theorem reads (Estep *et al.*, 2008*b*),

Theorem 1.69 *The errors satisfy*

$$\begin{aligned} (\psi, e) = {} & (f, \phi - \pi_V \phi) - a_1(u_h^{\{k\}}, \phi - \pi_V \phi_1) - c_1(u_h^{\{k\}}, u_h^{\{k\}}, \phi - \pi_V \phi) \\ & - b(\phi - \pi_V \phi, p_h) - d(T_{F,h}^{\{k\}}, \phi - \pi_V \phi) - b(u_h^{\{k\}}, z - \pi_Z z) \qquad (1.103) \\ & + (Q_F, \theta_F - \pi_{W_F} \theta_F) - a_2(T_{F,h}^{\{k\}}, \theta_F - \pi_{W_F} \theta_F) \\ & - c_2(u_h^{\{k\}}, T_{F,h}^{\{k\}}, \theta_F - \pi_{W_F} \theta_F) + (Q_S, \theta_S - \pi_{W_S} \theta_S) \\ & - a_3(T_{S,h}^{\{k\}}, \theta_S - \pi_{W_S} \theta_S) \qquad (1.104) \\ & + \left(T_{S,h}^{\{k\}} - \pi_S T_{F,h}^{\{k\}}, k_S (n \cdot \nabla \theta_S)\right)_{\Gamma_I} + \left(\pi_S T_{F,h}^{\{k\}} - T_{F,h}^{\{k\}}, k_S (n \cdot \nabla \theta_S)\right)_{\Gamma_I} \qquad (1.105) \\ & + \left(\sigma^{\{k\}}, \pi_{W_F} \theta_F - \pi_{W_S} \theta_S\right)_{\Gamma_I}. \qquad (1.106) \end{aligned}$$

Note the difference in (1.106) compared to (1.99); now there is only a projection error arising from a change of scale, without any transfer error expression.

We can prove that using the recovered flux recovers the expected cubic order of convergence, see (Estep *et al.*, 2008*b*).

Example 1.70 The recovered accuracy is easily demonstrated by considering the adapted meshes produced by (1.103)-(1.106). We repeat the computations in Ex. 1.67 using the modified error bound with the recovered flux derived from (1.103)-(1.106) to guide adaptive mesh refinement. We show the final adaptive meshes for the solid in Fig. 1.35. There is no mesh refinement near the boundaries, indicating that the flux error is no longer dominant.

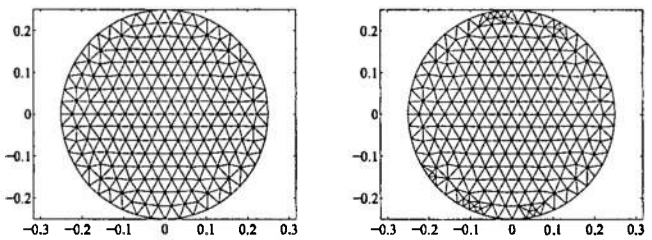


FIG. 1.35. Left: Final adaptive mesh in the solid when the quantity of interest is the temperature in a small region in the wake above the cylinder. Right: Final adaptive mesh in the solid when the quantity of interest is the temperature in a small region in the center of the solid.

1.7 The effect of iteration

In the presentation above, we have minimized the effects arising from the solution of nonlinear and/or fully coupled systems by carefully choosing the models and results that are discussed. Referring back to Fig. 1.3, we generally expect multiscale operator decomposition to require a number of iterations between the physical components. This raises additional issues that need to be addressed, e.g.

- The convergence of the iteration is always paramount. Note that the convergence is strongly affected by the fact that we are repeatedly upscaling and downscaling information. Indeed, this may even affect the definition of convergence, e.g. when coupling stochastic models to continuum models.
- When iteration is required, then we are passing information, along with error, between iteration levels as well as physical components, and this requires defining additional auxiliary quantities of interest and corresponding adjoint operators.
- The "single physics paradigm" means that in practice we may have access only to adjoint operators associated to the components, but not to the entire system. This has strong consequences on which contributions to the error may be estimated.

These issues are discussed in (Estep *et al.*, 2008*a*; Estep *et al.*, 2008*b*; Carey *et al.*, 2008*a*; Estep *et al.*, 2008*a*; Estep *et al.*, 2008*b*).

1.8 Conclusion

Multiphysics, multiscale models present significant challenges in terms of computing accurate solutions and for estimating the error in information computed from numerical solutions. In this chapter, we discuss the problem of computing accurate error estimates for one of the most common, and powerful, numerical approaches for multiphysics, multiscale problems called multiscale operator decomposition. This is a widely used technique for solving multiphysics, multiscale models. The general approach is to decompose the multiphysics and/or multiscale problem into components involving simpler physics over a relatively limited range of scales, and then to seek the solution of the entire system through some sort of iterative procedure involving numerical solutions of the individual components. In general, different components are solved with different numerical methods as well as with different scale discretizations. This approach is appealing because there is generally a good understanding of how to solve a broad spectrum of single physics problems accurately and efficiently, and because it provides an alternative to accommodating multiple scales in one discretization.

In the first part of this chapter, we describe the ingredients of adjoint-based *a posteriori* error analysis. We stress the need to accurately quantify stability of particular information to be computed from a model and the role of the adjoint problem for this purpose.

Turning to specific examples of multiscale, multiphysics models, we illustrate the general observation that the stability properties of such models are exceedingly complex. This heightens the importance of obtaining accurate information about stability.

We then describe how the techniques of *a posteriori* error analysis can be extended to multiscale operator decomposition solutions of multiphysics, multiscale problems. While the particulars of the analysis vary considerably with the problem, there are several key ideas underlying a general approach to treat operator decomposition multiscale methods, including:

- We identify auxiliary quantities of interest associated with information passed between physical components and solve auxiliary adjoint problems to estimate the error in those quantities.
- We deal with scale differences by introducing projections between discrete spaces used for component solutions and estimate the effects of those projections.
- The standard linearization argument used to define an adjoint operator associated with error analysis for a nonlinear problem may fail, requiring another approach to define adjoint operators.
- In this regard, the adjoint operator associated with a multiscale operator decomposition solution method is often different than the adjoint associ-

ated with the original problem, and the difference may have a significant impact on the stability of the method.

- In practice, solving the adjoint associated with the original fully-coupled problem may present the same kinds of multiphysics, multiscale challenges posed by the original problem, so attention must be paid to the solution of the adjoint problem.

We explain these ideas in the context of three specific examples.

REFERENCES

Atkinson, K. and Han, W. (2001). *Theoretical Numerical Analysis: A Functional Analysis Framework*. Springer.

Aubin, J. (2000). *Applied Functional Analysis*. John Wiley & Sons, Inc.

Bangerth, W. and Rannacher, R. (2003). *Adaptive Finite Element Methods for Differential Equations*. Birkhauser Verlag.

Barth, T. J. (2004). *A-Posteriori Error Estimation and Mesh Adaptivity for Finite Volume and Finite Element Methods*, Volume 41 of *Lecture Notes in Computational Science and Engineering*. Springer, New York.

Becker, R. and Rannacher, R. (2001). An optimal control approach to a posteriori error estimation in finite element methods. *Acta Numerica*, 1–102.

Cacuci, D. (1997). *Sensitivity and Uncertainty Analysis: Theory*, Volume I. Chapman & Hall/CRC.

Carey, G.F. (1982). Derivative calculation from finite element solutions. *Comp. Meth. in Applied Mech. and Engr.*, **35**, 1–14.

Carey, V., Estep, D., Larson, M., and Tavener, S. (2008*b*). Blockwise adaptivity for time dependent problems based on coarse scale adjoint solutions. in preparation.

Carey, V., Estep, D., and Tavener, S. (2006). A posteriori analysis and adaptive error control for operator decomposition methods for coupled elliptic systems I: One way coupled systems. *SINUM*. to appear.

Carey, V., Estep, D., and Tavener, S. (2008*a*). A posteriori analysis and adaptive error control for operator decomposition methods for elliptic systems II: Fully coupled systems. *In preparation*.

Cheney, W. (2000). *Analysis for Applied Mathematics*. John Wiley & Sons, Inc.

Delfour, M. and Dubeau, F. (1986). Discontinuous polynomial approximations in the theory of one-step, hybrid and multistep methods for nonlinear ordinary differential equations. *Math. Comp.*, **47**, 169–189.

Delfour, M., Hager, W., and Trochu, F. (1981). Discontinuous Galerkin methods for ordinary differential equations. *Math. Comp.*, **36**, 455–472.

Eriksson, K., Estep, D., Hansbo, P., and Johnson, C. (1995). Introduction to adaptive methods for differential equations. In *Acta numerica, 1995*, Acta Numer., pp. 105–158. Cambridge Univ. Press, Cambridge.

Eriksson, K., Estep, D., Hansbo, P., and Johnson, C. (1996). *Computational differential equations*. Cambridge University Press, Cambridge.

Eriksson, K., Johnson, C., and Thomée, V. (1985). Time discretization of parabolic problems by the Discontinuous Galerkin method. *RAIRO Modél. Math. Anal. Numér.*, **19**, 611–643.

Estep, D. (1995). A posteriori error bounds and global error control for ap-

proximation of ordinary differential equations. *SIAM J. Numer. Anal.*, **32**(1), 1–48.

Estep, D. and French, D. (1994). Global error control for the continuous Galerkin finite element method for ordinary differential equations. *RAIRO Modél. Math. Anal. Numér.*, **28**, 815–852.

Estep, D., Ginting, V., Shadid, J., and Tavener, S. (2008*a*). An a posteriori-a priori analysis of multiscale operator splitting. *SIAM J. Num. Analysis*, **46**, 1116–1146.

Estep, D., Ginting, V., Shadid, J., and Tavener, S. (2008*b*). A posteriori analysis of a multirate numerical method for ordinary differential equations. Submitted to *SIAM J. Num. Analysis*.

Estep, D., Holst, M., and Larson, M. (2005). Generalized Green's functions and the effective domain of influence. *SIAM J. Sci. Comput.*, **26**, 1314–1339.

Estep, D., Holst, M., and Mikulencak, D. (2002). Accounting for stability: a posteriori error estimates based on residuals and variational analysis. *Comm. Num. Meth. Engin.*, **18**, 15–30.

Estep, D. and Johnson, C. (1998). The computability of the Lorenz system. *Math. Models Meth. Appl. Sci.*, **8**, 1277–1305.

Estep, D., Larson, M. G., and Williams, R. D. (2000). Estimating the error of numerical solutions of systems of reaction-diffusion equations. *Mem. Amer. Math. Soc.*, **146**(696), viii+109.

Estep, D. and Larsson, S. (1993). The discontinuous Galerkin method for semilinear parabolic problems. *RAIRO Modél. Math. Anal. Numér.*, **27**, 35–54.

Estep, D. and Stuart, A. M. (2002). The dynamical behavior of the discontinuous Galerkin method and related difference schemes. *Math. Comp.*, **71**(239), 1075–1103 (electronic).

Estep, D., Tavener, S., and Wildey, T. (2008*a*). A posteriori analysis and improved accuracy for an operator decomposition solution of a conjugate heat transfer problem. *SINUM*, **46**, 2068–2089.

Estep, D., Tavener, S., and Wildey, T. (2008*b*). A posteriori error estimation and adaptive mesh refinement for a multi-discretization operator decomposition approach to fluid-solid heat transfer. *J. Comput. Phys.*, in revision.

Estep, D. and Williams, R. (1996). Accurate parallel integration of large sparse systems of differential equations. *Math. Models Meth. Appl. Sci.*, **6**, 535–568.

Folland, G. (1999). *Real Analysis*. John Wiley & Sons, Inc.

G.F. Carey, S.S. Chow, M.K. Seager (1985). Approximate boundary-flux calculations. *Comp. Meth. in Applied Mech. and Engr.*, **50**, 107–120.

Giles, M. and Süli, E. (2002). Adjoint methods for PDEs: A posteriori error analysis and postprocessing by duality. *Acta Numerica*, 145–236.

Higham, N. J. (2002). *Accuracy and Stability of Numerical Algorithms*. SIAM, Philadelphia.

Jamet, P. (1978). Galerkin–type approximations which are discontinuous in time for parabolic equations in a variable domain. *SIAM J. Numer. Anal.*, **15**, 912–928.

Lanczos, C. (1997). *Linear Differential Operators.* Dover Publications.

Lions, J. and Magenes, E. (1972). *Non-Homogeneous Boundary Value Problems and Applications*, Volume 1. Springer-Verlag, New York.

Lorenz, E. N. (1963). Deterministic non-periodic flows. *J. Atmos. Sci.*, **20**, 130–141.

Marchuk, G. I. (1995). *Adjoint equations and analysis of complex systems.* Kluwer.

Marchuk, G. I., Agoshkov, V. I., and Shutyaev, V. P. (1996). *Adjoint equations and perturbation algorithms in nonlinear problems.* CRC Press, Boca Raton, FL.

Paraschivoiu, M., Peraire, J., and Patera, A.T. (1997). A posteriori finite element bounds for linear functional output of elliptic partial differential equations. *Comput. Methods Appl. Mech. Engrg.*, **150**, 289–312.

Prigogine, I. and Lefever, R. (1968). Symmetry breaking instabilities in dissipative systems. *J. Chem. Phys*, **48**(4), 1695–1700.

Ropp, D. L. and Shadid, J. N. (2005). Stability of operator splitting methods for systems with indefinite operators: reaction-diffusion systems. *J. Comput. Phys.*, **203**(2), 449–466.

Sandelin, J. (2006). *Global Estimate and Control of Model, Numerical, and Parameter Error.* Ph. D. thesis, Department of Mathematics, Colorado State University, Fort Collins, CO 80523.

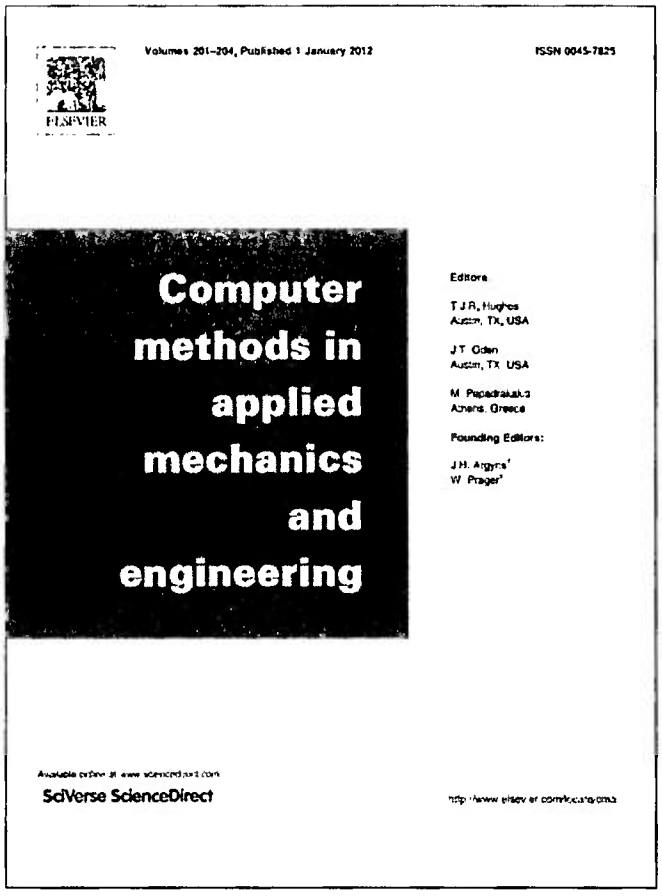
Schechter, M. (2002). *Real Analysis.* American Mathematical Society.

Thomée, V. (1980). *Galerkin Finite Element Methods for Parabolic Problems.* Springer-Verlag, New York.

Wheeler, M.F. (1974). A Galerkin procedure for estimating the flux for two-point boundary-value problems using continuous piecewise-polynomial spaces. *Numer. Math*, **2**, 99–109.

Wildey, T., Estep, D., and Tavener, S. (2008). *A posteriori* error estimation of approximate boundary fluxes. *Commun. Num. Meth. Engin.*, **24**, 421–434.

INDEX



(This is a sample cover image for this issue. The actual cover is not yet available at this time.)

Comput. Methods Appl. Mech. Engrg. 223–224 (2012) 10–27

Contents lists available at SciVerse ScienceDirect

Comput. Methods Appl. Mech. Engrg.

journal homepage: www.elsevier.com/locate/cma

A Posteriori analysis of a multirate numerical method for ordinary differential equations

D. Estep [a,*,1], V. Ginting [b,2], S. Tavener [c]

[a] Department of Statistics, Colorado State University, Fort Collins, CO 80523, United States
[b] Department of Mathematics, University of Wyoming, Laramie, WY 82071, United States
[c] Department of Mathematics, Colorado State University, Fort Collins, CO 80523, United States

ARTICLE INFO

Article history:
Received 7 September 2011
Accepted 24 February 2012
Available online 3 March 2012

Keywords:
Adjoint operator
A posteriori estimates
Discontinuous Galerkin method
Iterative method
Multirate method
Multiscale integration
Operator decomposition

ABSTRACT

In this paper, we analyze a multirate time integration method for systems of ordinary differential equations that present significantly different scales within the components of the model. The main purpose of this paper is to present a hybrid *a priori* – *a posteriori* error analysis that accounts for the effects of projections between the coarse and fine scale discretizations, the use of only a finite number of iterations in the iterative solution of the discrete equations, the numerical error arising in the solution of each component, and the effects on stability arising from the multirate solution. The hybrid estimate has the form of a computable *a posteriori* leading order expression and a provably-higher order *o priori* expression. We support this estimate by an *a priori* convergence analysis. We present several examples illustrating the accuracy of multirate integration schemes and the accuracy of the *o posteriori* estimate.

1. Introduction

In this paper, we analyze a multirate numerical method for a system of ordinary differential equation that presents significantly different scales for the rate of change of individual components of the model. A multirate method employs discretizations on significantly different time scales for different components of the problem. For simplicity, we consider a model that can be decomposed into two vector-valued components: find $y = (y_1, y_2)^T \in \mathbb{R}^n$ that satisfies

$$\begin{cases} \dot{y}_1 = F_1(y_1, y_2), & t \in (0, T], \\ \dot{y}_2 = F_2(y_1, y_2), & t \in (0, T], \\ y_1(0) = g_1, y_2(0) = g_2, \end{cases} \tag{1}$$

for a given initial condition $g = (g_1, g_2)^T$. Here, $y_i \in \mathbb{R}^{n_i}, i = 1, 2, n = n_1 + n_2$, and $F = (F_1, F_2)^T \in \mathbb{R}^n$, with $F_i(y) = F_i(y_1, y_2) \in \mathbb{R}^{n_i}$, $i = 1, 2$. If F_1 and F_2 induce significantly different rates of change in the respective solution components, then an heuristic consideration of accuracy suggests that it is most efficient to solve (1) using small time steps for the fast component and large time steps for the slow component. While we have assumed the form of (1) in which the slow and fast components are distinguished for the sake of exposition but we do not use knowledge of the slow and fast components in the analysis. Indeed, the estimates we obtain can be used to determine if a particular identification of fast and slow components is correct. Also, the method and analysis extend to systems with more than two scales in a straightforward way.

Such multiscale systems arise in a variety of applications, e.g. fusion and fission, reacting flows, circuit analysis, convection problems, and mechanical systems. As a useful example, we consider a discrete model consisting of a wire in a state of constant tension $\mathcal{T}$ supporting N masses, see Fig. 1. The masses $m_i, i = 1, \ldots, N$ have horizontal positions $x_i, i = 1, \ldots, N$ and vertical positions $y_i, i = 1, \ldots, N$. The horizontal spacing between masses is $a_r = x_r - x_{r-1}$, $r = 1, \ldots, N$. Applying Newton's second law and assuming the tensile forces are large compared with the gravitational forces, a linear damping model and that the masses are free to move in the vertical direction only, we have

$$m_r \frac{d^2 y_r}{dt^2} = -\mathcal{T} \sin(\theta_r) + \mathcal{T} \sin(\theta_{r+1}) - 2\gamma_r \frac{dy_r}{dt},$$

* Corresponding author.
E-mail address: estep@stat.colostate.edu (D. Estep).

[1] D. Estep's work is supported in part by the Defense Threat Reduction Agency (HDTRA1-09-1-0036), Department of Energy (DE-FG02-04ER25620, DE-FG02-05ER25699, DE-FC02-07ERS4909, DE-SC0001724, DE-SC0005304), Lawrence Livermore National Laboratory (8573139, B584647, B59049S), the National Aeronautics and Space Administration (NNG04GH63G), the National Institutes of Health (SR01GM096192-02), the National Science Foundation (DMS-0107832, DMS-0715135, DGE-0221595003, MSPA-CSE-0434354, ECCS-0700559, DMS-1016268, DMS-FRG-1065046), Idaho National Laboratory (00069249, 00115474), and the Sandia Corporation (P0299784).

[2] V. Ginting's work is supported in part by the National Science Foundation (DMS-1016283), the Department of Energy (DE-FE0004832 and DE-SC0004982), and the Center for Fundamentals of Subsurface Flow of the School of Energy Resources of the University of Wyoming (WYDEQ49811GNTG and WYDEQ49811PER).

0045-7825/$ - see front matter © 2012 Elsevier B.V. All rights reserved.
doi:10.1016/j.cma.2012.02.021

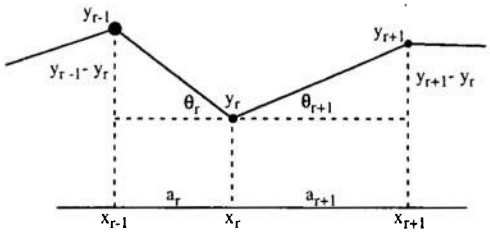


Fig. 1. A model of masses connected by a wire under tension.

where θ_r and θ_{r+1} denote the angles (measured positive counterclockwise) between the horizontal and the wires connecting the masses m_{r-1} and m_r, and m_r and m_{r+1} respectively as illustrated.

Without loss of generality we assume that $\mathcal{T} = 1$ and $y_0 = y_{N+1} = 0$. To create a system with two time scales, we consider two heavy masses coupled to $(N-2)$ lighter masses with $m_0 = m_1 = M \gg m_2 = \cdots = m_N = m$, $a_1 = a_2 = A \gg a_3 = \cdots = a_N = a_{N+1} = a$, see Fig. 2(a), and define $\Gamma = \gamma_1 M^{-1} = \gamma_2 M^{-1}$ and $\gamma = \gamma_2 m^{-1} = \cdots \gamma_N m^{-1}$. Making the small angle approximation $\sin(\theta) \approx \theta \approx \tan(\theta)$and introducing $v_r = \frac{dy_r}{dt}$, $r = 1,\ldots,N$, and $v_{N+r} = y_r$, $r = 1,\ldots,N$, we rewrite the system as a $2N$-dimensional system of first-order differential equations

$$\frac{dv}{dt} = \begin{pmatrix} \mathbf{D} & \mathbf{A} \\ \mathbf{I}_{N\times N} & \mathbf{0}_{N\times N} \end{pmatrix} v, \tag{2}$$

where

$$\mathbf{A} = \begin{pmatrix} -2(AM)^{-1} & (AM)^{-1} & 0 & 0 & & 0 \\ (AM)^{-1} & -(AM)^{-1} - (aM)^{-1} & (aM)^{-1} & 0 & & 0 \\ 0 & (am)^{-1} & -2(am)^{-1} & (am)^{-1} & & 0 \\ \vdots & & \ddots & \ddots & \ddots & \vdots \\ 0 & & & (am)^{-1} & -2(am)^{-1} & (am)^{-1} \\ 0 & & & & (am)^{-1} & -2(am)^{-1} \end{pmatrix},$$

$$\mathbf{D} = \begin{pmatrix} -2\Gamma \mathbf{I}_{2\times 2} & 0 \\ 0 & -2\gamma \mathbf{I}_{N-2\times N-2} \end{pmatrix},$$

and $\mathbf{I}_{n\times n}$ denotes the $n \times n$ identity matrix and $\mathbf{0}_{n\times n}$ is the $n \times n$ zero matrix. We set $N = 7$, $A = .25$, $a = .1$, $M = 10$, and $m = .01$ and plot a typical solution in Fig. 2(b). The two scales for evolution are clear. We also show the pointwise accuracy for several multirate numerical solutions in Fig. 2(c). The accuracy gained by using smaller time steps for the fine scale is displayed, as is the fact that there is a limit to the accuracy that can be gained. Indeed, using a multirate approach affects both accuracy and stability. Consider the case of a two-dimensional system (1) with scalar fast y_1 and slow y_2 components. In this case, we can write

$$\frac{dy_1}{dy_2} = \frac{f_1}{f_2}(y_1, y_2)$$

and solving (1) amounts to tracing out a smooth curve with slope f_1/f_2 at each point. Using a multirate method for approximating the change $(y_1, y_2) \to (y_1 + \Delta y_1, y_2 + \Delta y_2)$ amounts to replacing the implicit slope $\frac{f_1}{f_2}(y_1 + \Delta y_1, y_2 + \Delta y_2)$ at the new point by $\frac{f_1}{f_2}(y_1 + \Delta y_1, y_2)$, which has been "frozen" at the previous y_2 value. Drawing a few examples provides convincing evidence that this affects both accuracy and stability regardless of how well we approximate the step between (y_1, y_2) and $(y_1 + \Delta y_1, y_2 + \Delta y_2)$.

There is a significant literature on multirate numerical methods, see for example [48,29,1,45,30,55,56,50,2,37,8,3,33,14,19,18,44,15,38,32,5,51,36,40,41,46,11,49,57,42,60,13,58,61,54,52,53,34,10]. By and large, these references are focused on application and standard *a priori* analysis issues, e.g. stability, accuracy, and convergence properties. Such analysis is not generally useful for estimating the error in specific numerical solutions. The main goal of this paper is to derive a computable *a posteriori* error representation that *accurately* estimates the error in a specified quantity of interest computed from a multirate solution of (1).

Our analysis adapts a well developed approach based on duality, adjoint operators, and variational analysis, see for example in [43,39,22,20,21,26,7,31,4,6]. In order to use a variational framework for analysis, we represent the numerical method as a finite element method. In this paper, we consider the so-called discontinuous Galerkin (dG) method [16,17,59,22] which employs piecewise polynomial shape functions. We can recover many standard finite difference schemes by appropriate choice of quadrature applied to the integrals defining the finite element solution. Our analysis also applies to the so-called continuous Galerkin (cG) method, which yields other families of difference schemes upon application of quadrature [24].

Our approach is based on the observation that a multirate method shares some features of multiscale operator decomposition methods for multiscale problems [43,9,27,28,25,23]. In particular, the need to project the approximate solutions between the discretizations at different scales and the practical use of incomplete iteration when solving the discrete equations has effects on accuracy and stability similar to those caused by operator decomposition. Indeed, we adapt ideas used in the investigation of operator splitting for reaction-diffusion equations [25] to carry out the analysis. In particular, we define adjoint operators for both the original nonlinear operator and the multirate-discretization operator independently by linearization with respect to a solution that is assumed to be common to both operators and obtain analogs of the standard Green's formula representing the quantity of interest relative to the common solution. We then carry out the *a posteriori* error analysis by comparing the resulting linear expressions. The fact that multiscale operator decomposition affects stability is directly reflected in the form of the estimate, which involves the difference between certain adjoint operators associated with the solution operator for the original problem and the multirate – operator decomposition solution operator. A practical difficulty with such

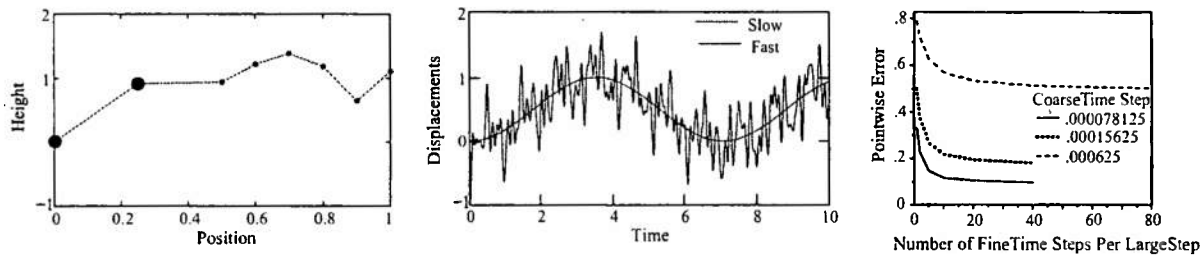


Fig. 2. Typical simulations of the two scale system. Left: A solution at a fixed time. Center: The components at y_1 (slow) and y_7 (fast) versus time. Right: Plots of pointwise accuracy versus the fine scale time step for three coarse scale time steps.

estimates is the fact that it is unlikely that the adjoint for the original problem is available since it poses the same multiscale challenge as solving the original forward problem. Therefore, we derive an estimate expressed as a computable leading order expression obtained using *a posteriori* arguments with remainder that cannot be estimated but which is provably higher order. We call this a hybrid o *priori* – *a posteriori* estimate.

Our analysis shares some aspects of the results in the penetrating series of papers [40–42] by Logg. In those papers, he carries out an o *priori* convergence analysis, an *a posteriori* error analysis, analyses the convergence of a fixed point scheme for the numerical method, and constructs an adaptive time stepping algorithm for what is termed multiadaptive cG and dG methods. These methods generalize the notion of multirate schemes to allow different time steps for *each* individual component of a system of differential equations of size N.

Our results differ from this work in two significant ways. First, Logg does not consider the effects of projecting between the discretizations at different scales on the accuracy of the approximation. If projections are not employed, then the quadrature used in the finite element formulation should be carried out on the fastest time scale in order to avoid integrating non-smooth discrete approximations. However, this greatly increases the expense of a multirate scheme. In practice, a projections are introduced in a de facto fashion that is rarely arranged on the finest time scale, which potentially has a significant effect on accuracy. In any case, our analysis provides the means to quantify the effect of the disparate time scales in the discretizations on overall accuracy.

Secondly, Logg does not consider the effect of incomplete iteration in the solution of the discrete equations. That is, the complete convergence of the discrete system equations that must be solved at each "synchronization time" is assumed. In practice, it is common to carry out only a few or fixed number of iterations (perhaps only 1!), and the fewer iterations that are used, the more the discretization scheme behaves like an operator decomposition method. Our analysis of the effect of incomplete iteration provides the means to strike a balance between the discretization and iteration errors.

The rest of the paper is organized as follows. In Section 2, we formulate multirate Galerkin finite element methods for (1). We present some o *priori* convergence results in Section 3. The main results on the o *posteriori* analysis of the multirate method are presented in Section 4 followed by several numerical examples in Section 5. In Section 6, we give the details of proof of the *a priori* result. Finally, in Section 7, we present a conclusion.

2. A multirate Galerkin finite element method

2.1. A fully implicit multirate Golerkin finite method

We discretize $[0,T]$ into $0 = t_0 < t_1 < t_2 < \cdots < t_N = T$ with time steps $\Delta t_n = t_n - t_{n-1}$, $\Delta t = \max_{1\leqslant n\leqslant N}\{\Delta t_n\}$, and time intervals $I_n = (t_{n-1}, t_n)$. Let $L_{i,n}$, $i = 1, 2$ be two positive integers, where $L_{1,n}$ denotes the number of time steps used to solve the fast subsystem and $L_{2,n}$ the number of steps used for the slow subsystem. We denote time steps for each component in the Galerkin formulation by $\Delta s_{i,n} = \Delta t_n/L_{i,n}$, with $\Delta s_i = \max_{1\leqslant n\leqslant N}\{\Delta s_{i,n}\}$. Within I_n we set $I_{l,n} = (t_{l-1,n}, t_{l,n})$ with $t_{l,n} = t_{n-1} + l\Delta s_{1,n}$, for $l = 0, 1, 2, \ldots, L_{1,n}$, and $I_{k,n} = (t_{k-1,n}, t_{k,n})$ with $t_{k,n} = t_{n-1} + k\Delta s_{2,n}$, for $k = 0, 1, 2, \ldots, L_{2,n}$.

We use an extension of the discontinuous Galerkin method [26], which is an appropriate method for dissipative problems. The analysis naturally extends to the continuous Galerkin method, which is more appropriate for problems with conserved quantities [26]. The finite element approximate solutions are sought in piecewise polynomial spaces,

$$\mathcal{V}^{(q_1)}(I_n) = \{U : U|_{I_{l,n}} \in \mathcal{P}^{(q_1)}(I_{l,n}), 1 \leqslant l \leqslant L_{1,n}\},$$
$$\mathcal{V}^{(q_2)}(I_n) = \{U : U|_{I_{k,n}} \in \mathcal{P}^{(q_2)}(I_{k,n}), 1 \leqslant k \leqslant L_{2,n}\}.$$

The multirate Galerkin finite element solution is to find $Y = (Y_1\ Y_2)^T$ with $Y_1|_{I_n} \in \mathcal{V}^{(q_1)}(I_n)$ and $Y_2|_{I_n} \in \mathcal{V}^{(q_2)}(I_n)$ satisfying

$$\int_{I_{l,n}} (\dot{Y}_1 - F_1(Y_1, Y_2), V)\,dt + \left([Y_1]_{l-1,n}, V^+_{l-1}\right) = 0, \quad \forall V \in \mathcal{P}^{(q_1)}(I_{l,n}), \tag{3}$$

for $l = 1, 2, \ldots, L_{1,n}$, and

$$\int_{I_{k,n}} (\dot{Y}_2 - F_2(\Pi Y_1, Y_2), W)\,dt + \left([Y_2]_{k-1,n}, W^+_{k-1}\right) = 0, \quad \forall W \in \mathcal{P}^{(q_2)}(I_{k,n}), \tag{4}$$

for $k = 1, 2, \ldots, L_{2,n}$. Here Π is a projection operator between the two different meshes. In the simplest case, $\Pi = I$ implies that the integration in (4) is carried out on the finest mesh. As noted in [9] this projection can become a significant source of error whenever information is transferred between two different discretizations and its effects should be estimated separately as a potentially significant component of the total error. Eqs. (3) and (4) together comprise a system of size $L_{1,n}(q_1 + 1) + L_{2,n}(q_2 + 1)$ which is solved implicitly.

We note that many standard finite difference schemes can be obtained by applying appropriate quadrature formulas to the integrals defining the finite element approximation. A straightforward modification of the analysis below extends the results to such difference schemes.

2.2. An iterative multirate Galerkin finite element method

Without loss of generality, we assume $L_{1,n} = d_n L_{2,n}$ for some positive integer d_n, i.e., $L_{1,n}$ is divisible by $L_{2,n}$. The iterative multirate Galerkin finite element is to create a sequence $Y^{(m)} = \left(Y_1^{(m)}\ Y_2^{(m)}\right)^T$ with $Y_1^{(m)}|_{I_n} \in \mathcal{V}^{(q_1)}(I_n)$ and $Y_2^{(m)}|_{I_n} \in \mathcal{V}^{(q_2)}(I_n)$ determined by Algorithm 1.

Algorithm 1: Iterative multirate Galerkin finite element method

for $n = 1$ to N **do**
  Set $Y_2^{(0)} = Y_2^{(M_{n-1})}(t_{n-1}^-)$
  for $m = 1$ to M_n **do**
    Set $Y^{(m)}(t_{n-1}^-) = Y^{(M_{n-1})}(t_{n-1}^-)$.
    for $l = 1$ to $L_{1,n}$ **do**
      Compute $Y_1^{(m)}(t)$ for $t \in I_{l,n}$ satisfying

$$\int_{I_{l,n}} \left(\dot{Y}_1^{(m)} - F_1(Y_1^{(m)}, Y_2^{(m-1)}), V\right)dt + \left([Y_1^{(m)}]_{l-1,n}, V^+_{l-1}\right) = 0, \tag{5}$$

      for all $V \in \mathcal{P}^{(q_1)}(I_{l,n})$.
    end for
    for $k = 1$ to $L_{2,n}$ **do**
      Compute $Y_2^{(m)}(t)$ satisfying

$$\int_{I_{k,n}} \left(\dot{Y}_2^{(m)} - F_2(\Pi Y_1^{(m)}, Y_2^{(m)}), W\right)dt + \left([Y_2^{(m)}]_{k-1,n}, W^+_{k-1}\right) = 0, \tag{6}$$

      for all $W \in \mathcal{P}^{(q_2)}(I_{k,n})$.
    end for
  end for
end for

3. *A priori* theory

Under appropriate assumptions, the solution of (1) exists and is unique in $[0,T]$, see [12,47]. An *a priori* analysis for a multi-adaptive Galerkin solution of (3), (4) is provided in [42]. For completeness, we present a short *a priori* convergence analysis that is required for the hybrid *a posteriori-a priori* estimate for the iterative approximation produced by Algorithm 1.

3.1. An analytic iterative method

As a tool for analysis, we consider a theoretical approximation obtained by iterating analytic solutions of the fast and slow components. Specifically, we let $y^{(m)} = \left(y_1^{(m)}\ y_2^{(m)}\right)^T$ denote the analytic approximation of (1) at iteration level m obtained using the iterative procedure defined in Algorithm 2.

Algorithm 2: Analytic fixed point iteration

for $n = 1$ to N **do**
  Set $y_2^{(0)} = y_2^{(M_{n-1})}(t_{n-1})$
  for $m = 1$ to M_n **do**
    Compute $y_1^{(m)}(t)$ for $t \in I_n$ satisfying

$$\begin{cases} \dot{y}_1^{(m)} = F_1\left(y_1^{(m)}, y_2^{(m-1)}\right) \\ y_1^{(m)}(t_{n-1}) = y_1^{(M_{n-1})}(t_{n-1}) \end{cases} \tag{7}$$

    Compute $y_2^{(m)}(t)$ for $t \in I_n$ satisfying

$$\begin{cases} \dot{y}_2^{(m)} = F_2\left(y_1^{(m)}, y_2^{(m)}\right) \\ y_2^{(m)}(t_{n-1}) = y_2^{(M_{n-1})}(t_{n-1}). \end{cases} \tag{8}$$

  end for
end for

We begin the *a priori* analysis by analyzing the convergence of the method in Algorithm 2 over $I_n = [t_{n-1}, t_n]$. The analysis is focused on determining a time interval I_n over which the iteration converges to the correct solution. For this purpose, we assume that the solution from previous time level t_{n-1} has been obtained exactly, i.e., $y^{(m)}(t_{n-1}) = y(t_{n-1})$. Following the classic method of successive approximations, we integrate (7) and (8) to get

$$\begin{aligned} y_1^{(m)}(t) &= y_1(t_{n-1}) + \int_{t_{n-1}}^{t} F_1\left(y_1^{(m)}, y_2^{(m-1)}\right) ds, \\ y_2^{(m)}(t) &= y_2(t_{n-1}) + \int_{t_{n-1}}^{t} F_2\left(y_1^{(m)}, y_2^{(m)}\right) ds, \end{aligned} \tag{9}$$

Any continuous functions satisfying (9) also satisfies (7) and (8). Hence, the goal is to first show that each of the integral equations in (9) has a solution for a fixed m. We then proceed to show that the iteration solving (9) (and thus (7) and (8)) converges to the exact solution of (1). Details of this investigation are presented in Section 6. The main result is the following theorem.

Theorem 3.1. *There exists a time $t_n > t_{n-1}$ such that the sequence of functions $\left\{y_1^{(m)}\right\}$ and $\left\{y_2^{(m)}\right\}$ produced by the integral Eq. (9) converge to the exact solution of (1) on time interval $I_n = [t_{n-1}, t_n]$.*

3.2. Convergence of the iterative multirate Galerkin finite element method

We now turn to the *a priori* error analysis of the numerical solution described in Algorithm 1. An unusual feature of this problem is the numerical solution involves an alternating sequence of $Y_1^{(m)}$ and $Y_2^{(m)}$, which result from consistent numerical discretizations of (7) and (8) respectively. The analysis is carried out using the analog of the standard local error analysis for a finite difference scheme. For each component solution on each interval, we decompose the error into a sum of the error in the initial condition inherited from the previous component solution and the error of the numerical solution of the component assuming exact initial conditions on the current interval. We describe the main result below and give the detailed proof in Section 6.

Theorem 3.2. *Let $y_1^{(m)}$ and $y_2^{(m)}$ be the solution of anolytic fixed point iteration governed by (7) and (8), respectively, and $Y_1^{(m)}$ and $Y_2^{(m)}$ be their dG finite element opproximation with Π= the identity. Then the finite element error ot the finol time for $q_1 = 0$, 1 ond $q_2 = 0$, 1 is bounded, and*

$$|e^{(M_N)}(t_N^-)| \approx \mathbf{O}\left(\Delta s_1^{q_1+1}\right) + \mathbf{O}\left(\Delta s_2^{q_2+1}\right) + \mathbf{O}(|r_2|),$$

where $r_2 = \max_{1\le n\le N}\left|r_2^{(M_n)}\right|_{I_n}$, and $\left|r_2^{(m)}\right|_{I_n}$ is the iterotion residuol written in Lemmo 6.3.

Notice that there is a term quantifying the iteration residual, namely $r_2^{(m)}$. Provided the sequence of solutions are driven to produce small iteration residual relative to the errors produced by the finite element solution, then this term is negligible.

4. *A posteriori* analysis

It turns out that is feasible to treat the effects of projecting between the discretizations at different scales and the use of finite iterations in the iterative solution of the discrete equations separately, and doing so greatly simplifies notation.

4.1. Anolysis for the implicit multirote method

To define the adjoint, we set $z = s\,y + (1-s)Y$, with $s \in [0,1]$ and then let $\overline{F'(z)}$ be a matrix whose entries are

$$\overline{F'(z)}_{ij} = \int_0^1 \frac{\partial F_i}{\partial y_j}(z) ds.$$

Consequently, $F(y) - F(Y) = \overline{F'(z)}(y - Y)$. We note that

$$-\dot{e} + \overline{F'(z)}e = (-\dot{y} + F(y)) + (\dot{Y} - F(Y)) = \dot{Y} - F(Y).$$

Furthermore, using continuity of y,

$$\begin{aligned} e_{i-1,n}^+ &= y_{i-1,n}^+ - Y_{i-1,n}^+ = \left(y_{i-1,n}^- - Y_{i-1,n}^-\right) - \left(Y_{i-1,n}^+ - Y_{i-1,n}^-\right) \\ &= e_{i-1,n}^- - [Y]_{i-1,n}. \end{aligned}$$

Associated with the finite element solution, we denote by ϑ the generalized Green's function that satisfies an adjoint problem

$$\begin{cases} -\dot{\vartheta} = \overline{F'(z)}^T\vartheta, \quad t_n > t \geqslant t_{n-1} \\ \vartheta(t_n) = \psi_n, \end{cases} \tag{10}$$

On time interval $I_{i,n}$, $i = 1,2,\ldots,L_{1,n}$,

$$\begin{aligned} 0 &= \int_{I_{i,n}} \left(e, \dot{\vartheta} + \overline{F'(z)}^T\vartheta\right) dt \\ &= \left(e_{i,n}^-, \vartheta_{i,n}\right) - \left(e_{i-1,n}^+, \vartheta_{i-1,n}\right) + \int_{I_{i,n}} \left(-\dot{e} + \overline{F'(z)}e, \vartheta\right) dt. \end{aligned} \tag{11}$$

Combining all these expressions yields the recursive relation

$$\left(e_{l,n}^-, \vartheta_{l,n}\right) = \left(e_{l-1,n}^-, \vartheta_{l-1,n}\right) - \int_{I_{l,n}} (\dot{Y} - F(Y), \vartheta) dt - \left([Y]_{l-1,n}, \vartheta_{l-1,n}\right) \tag{12}$$

Theorem 4.1. *The implicit multirate Galerkin finite element solution sotisfies the error equation over one time step I_n:*

$$\left(e_n^-, \psi_n\right) = \left(e_{n-1}^-, \vartheta_{n-1}\right) + \mathcal{Q}_{1,n} + \mathcal{Q}_{2,n} + \mathcal{Q}_{II,n}$$

where

$$\mathcal{Q}_{1,n} = -\sum_{l=1}^{L_{1,n}} \left\{ \int_{I_{l,n}} \left(\dot{Y}_1 - F_1(Y_1, Y_2), \vartheta_1\right) dt + \left([Y_1]_{l-1,n}, \vartheta_{1,l-1,n}\right) \right\},$$

$$\mathcal{Q}_{2,n} = -\sum_{k=1}^{L_{2,n}} \left\{ \int_{I_{k,n}} \left(\dot{Y}_2 - F_2(\Pi Y_1, Y_2), \vartheta_2\right) dt + \left([Y_2]_{k-1,n}, \vartheta_{2,k-1,n}\right) \right\},$$

$$\mathcal{Q}_{II,n} = \sum_{k=1}^{L_{2,n}} \int_{I_{k,n}} (F_2(Y_1, Y_2) - F_2(\Pi Y_1, Y_2), \vartheta_2)\, dt.$$

Furthermore, by setting $\psi_N = \psi$ for o given ψ ond $\psi_{n-1} = \vartheta_{n-1}$ for $n = N, N-1, \ldots, 2$, the error of implicit multirate Galerkin finite element method at final time $t_N = T$ can be expressed as

$$(e_N^-, \psi) = \sum_{n=1}^{N} (\mathcal{Q}_{1,n} + \mathcal{Q}_{2,n} + \mathcal{Q}_{II,n}). \tag{13}$$

The term $\mathcal{Q}_{1,n}$ represents the finite element residual associated with the fast time scale subsystem, while $\mathcal{Q}_{2,n}$ represents the finite element residual associated with the slow time scale. The term $\mathcal{Q}_{II,n}$ represents the effect of projection of Y_1 from the fast to the slow time scale. We use $\mathcal{Q}_{1,n}, \mathcal{Q}_{2,n}, \mathcal{Q}_{II,n}$ to distinguish these terms from the closely related but distinct terms $Q_{1,n}, Q_{2,n}, Q_{II,n}$ appearing in Theorems 4.2, 4.3, and 4.4 below.

4.2. Anolysis for the iterative multirote method

To simplify notation, we now assume that the projection Π= the identity. As discussed above, a key feature of the analysis is the realization that the analytic fixed point iteration is naturally associated with a different adjoint operator than the original problem. Our approach [25] to overcome this issue is to use a different linearization than commonly used for nonlinear problems. We assume that the operators for the original problem and the analytic fixed point iteration share a common solution, and use that as a linearization point. The simplest example of such a solution is a steady-state solution, which can be guaranteed to exist by assuming homogeneity in the right-hand side, i.e.,

$$F(0) = 0.$$

This is generally not restrictive in practice, but this assumption can be generalized (see [25]). We let

$$\overline{F'_{ij}(y)} = \int_0^1 \frac{\partial F_i}{\partial y_j}(sy) ds, \quad i,j = 1,2, \tag{14}$$

and $\overline{F'}$ denotes the square matrix whose entries are (14). Then $F(y) = \overline{F'(y)}y$ and so $\dot{y} = \overline{F'(y)}y$. Associated with this linearized form, we denote by φ the generalized Green's function satisfying the following adjoint problem:

$$\begin{cases} -\dot{\varphi} = \overline{F'(y)}^{\top}\varphi, & t \in (T, 0], \\ \varphi(T) = \psi. \end{cases} \tag{15}$$

On subinterval $I_n = (t_{n-1}, t_n)$, we define the solution operators Φ_n associated with the Green's function,

$$\varphi(t) = \Phi_n(t)\psi_n,$$

for $t_n > t \geqslant t_{n-1}$ and some initial data ψ_n. We can obtain a solution representation using the Green's functions, by multiplying y with the adjoint Eq. (15) and integrating in time, integrating by parts,

$$(y_n, \psi_n) = (y_{n-1}, \Phi_n(t_{n-1})\psi_n). \tag{16}$$

4.2.1. Anolysis of the onalytic fixed point iterotion

To simplify presentation, we express the analytic fixed point iteration in Algorithm 2 in a more compact format. In particular, for any iteration index m, we write (7) and (8) as

$$\dot{y}^{(m)} = F(y^{(m)}) + \delta_y^{(m)}, \tag{17}$$

where

$$\delta_y^{(m)} = -\left[F_1\left(y_1^{(m)}, y_2^{(m)}\right) - F_1\left(y_1^{(m)}, y_2^{(m-1)}\right)0\right]^{\top}. \tag{18}$$

The vector $\delta_y^{(m)}$ can be interpreted as a residual at the iteration level m.

To define an adjoint for the analytic fixed point iteration in Algorithm 2, we let $\varphi_i^{(k)}$ denote the generalized Green's function that satisfies an adjoint problem on time interval I_n as given in Algorithm 3.

Algorithm 3: Adjoint for the analytic fixed point iteration

Set $\varphi_1^{(0)} = \psi_{1,n}$
for $k = 1$ to K_n **do**
Compute $\varphi_2^{(k)}$ satisfying

$$\begin{cases} -\dot{\varphi}_2^{(k)} = \overline{F'_{22}(y^{(m)})}^{\top}\varphi_2^{(k)} + \overline{F'_{12}(y^{(m)})}^{\top}\varphi_1^{(k-1)}, & t_n > t \geqslant t_{n-1} \\ \varphi_2^{(k)}(t_n) = \psi_{2,n}, \end{cases} \tag{19}$$

Compute $\varphi_1^{(k)}$ satisfying

$$\begin{cases} -\dot{\varphi}_1^{(k)} = \overline{F'_{11}(y^{(m)})}^{\top}\varphi_1^{(k)} + \overline{F'_{21}(y^{(m)})}^{\top}\varphi_2^{(k)}, & t_n > t \geqslant t_{n-1} \\ \varphi_1^{(k)}(t_n) = \psi_{1,n}, \end{cases} \tag{20}$$

end for

Notice that the adjoint problems are solved backward in time and in the reverse order to that of the forward problem, starting with $\varphi_2^{(k)}$ followed by $\varphi_1^{(k)}$. These generalized Green's functions are an iterative approximation of (15). As in the forward problem, we can also rewrite this last algorithm into a compact form

$$-\dot{\varphi}^{(k)} = \overline{F'(y^{(m)})}^{\top}\varphi^{(k)} + \xi^{(k)}, \tag{21}$$

for adjoint iteration level k. Here

$$\xi^{(k)} = -\left[0, \overline{F'_{12}(y^{(m)})}^{\top}\left(\varphi_1^{(k)} - \varphi_1^{(k-1)}\right)\right]^{\top},$$

is the residual of the adjoint at iteration level k. We also introduce the solution operators $\Phi_n^{(k)}$, with $\varphi^{(k)}(t) = \Phi_n^{(k)}(t)\psi_n$, for $t_n > t \geqslant t_{n-1}$. To get a representation of the iterative implicit solution, we follow a similar derivation as for the forward problem. Multiplying $y^{(m)}$ with (21), integrating each over I_n, and applying integration by parts, we obtain

$$\left(y_n^{(m)}, \psi_n\right) = \left(y_{n-1}^{(m)}, \Phi_n^{(k)}(t_{n-1})\psi_n\right) - \int_{I_n} \left(-\dot{y}^{(m)} + \overline{F'(y^{(m)})}y^{(m)}, \varphi^{(k)}\right) dt - \int_{I_n} \left(y^{(m)}, \xi^{(k)}\right) dt.$$

Using (17), we obtain the solution representation of the analytic iterative implicit method

$$(y_n^{(m)},\psi_n)=\left(y_{n-1}^{(m)},\Phi_n^{(k)}(t_{n-1})\psi_n\right)+\int_{I_n}\left(\dot\delta_y^{(m)},\varphi^{(k)}\right)dt-\int_{I_n}\left(y^{(m)},\xi^{(k)}\right)dt. \tag{22}$$

We note that this representation is not in the standard format (in which the solution at the current time level solely depends on the previous time level values). It contains artifacts arising from the iterative procedure used to compute both forward and backward problems. The second term can be interpreted as the weighted average of the forward problem residual over a time step. The third term, on the other hand, is the weighted average of the backward problem residual over a time step. Thus, the iterative nature of solution procedure is reflected in this representation. Once convergence is reached both on forward and backward problems, then the standard convention of solution representation using the adjoint technique is recovered.

We are now able to express the error representation of the iterative implicit method. First, we state a lemma concerning an error equation over one time step.

Lemma 4.1. *The analytic fixed paint iteratian satisfies the following errar equatian aver ane time step:*

$$(y_n-y_n^{(m)},\psi_n)=\left(y_{n-1},\Delta\Phi_n^{(k)}(t_{n-1})\psi_n\right)-\int_{I_n}\left(\delta_y^{(m)},\varphi^{(k)}\right)dt+\int_{I_n}\left(y^{(m)},\xi^{(k)}\right)dt$$

where $\Delta\Phi_n^{(k)}=\Phi_n-\Phi_n^{(k)}$.

Proof. This lemma is derived by subtracting (22) from (16) and setting $y_{n-1}^{(m)}=y_{n-1}$. □

Note that there are terms that are not computable in this expression. The term $\Delta\Phi_n^{(k)}$ is definitely not computable, though when convergence in the adjoint computation is reached, this term vanishes. Nevertheless, in the context of finite number of iterations, we desire to quantify $\Delta\Phi_n^{(k)}$. This is made more precise below.

4.2.2. Analysis of the iterative multirate Galerkin finite element method

To setup the adjoint, let $z^{(m)}=sy^{(m)}+(1-s)Y^{(m)}$, with $s\in[0,1]$. Then let $\overline{F'(z^{(m)})}$ be a matrix whose entries are

$$\overline{F'(z^{(m)})}_{ij}=\int_0^1\frac{\partial F_i}{\partial y_j}\left(z^{(m)}\right)ds.$$

Consequently, $F(y^{(m)})-F(Y^{(m)})=\overline{F'(z^m)}(y^{(m)}-Y^{(m)})$. Associated with the finite element solution, we denote by $\vartheta^{(k)}=\left[\vartheta_1^{(k)}\vartheta_2^{(k)}\right]^\top$, a sequence of generalized Green's function that satisfies an adjoint problem

Algorithm 4: Adjoint for the iterative multirate Galerkin finite element method

Set $\vartheta_1^{(0)}=\psi_{1,n}$
for $k=1$ to K_n **do**
Compute $\vartheta_2^{(k)}$ satisfying

$$\begin{cases}-\dot\vartheta_2^{(k)}=\overline{F'_{22}(z^{(m)})}^\top\vartheta_2^{(k)}+\overline{F'_{12}(z^{(m)})}^\top\vartheta_1^{(k-1)}, & t_n>t\geqslant t_{n-1}\\ \vartheta_2^{(k)}(t_n)=\psi_{2,n},\end{cases} \tag{23}$$

Compute $\vartheta_1^{(k)}$ satisfying

$$\begin{cases}-\dot\vartheta_1^{(k)}=\overline{F'_{11}(z^{(m)})}^\top\vartheta_1^{(k)}+\overline{F'_{21}(z^{(m)})}^\top\vartheta_2^{(k)} & t_n>t\geqslant t_{n-1}\\ \vartheta_1^{(k)}(t_n)=\psi_{1,n},\end{cases} \tag{24}$$

end for

As was the case in the adjoint formulation associated with analytic fixed point iteration, this algorithm can be expressed as a compact form

$$-\dot\vartheta^{(k)}=\overline{F'(z^{(m)})}^\top\vartheta^{(k)}+\eta^{(k)}, \tag{25}$$

where

$$\eta^{(k)}=-\left[0\overline{F'_{12}(z^{(m)})}^\top\left(\vartheta_1^{(k)}-\vartheta_1^{(k-1)}\right)\right]^\top,$$

is the residual of the adjoint at iteration level k.

At this stage, we are in position to derive an error equation associated with the numerical discretization of the analytic fixed point iteration. Let $e^{(m)}=y^{(m)}-Y^{(m)}$. On time interval $I_{l,n}$, $l=1,2,\ldots,L_{1,n}$,

$$\begin{aligned}0&=\int_{I_{l,n}}\left(e^{(m)},\dot\vartheta^{(k)}+\overline{F'(z^{(m)})}^\top\vartheta^{(k)}+\eta^{(k)}\right)dt\\&=\left(e_{l,n}^{(m)-},\vartheta_{l,n}^{(k)}\right)-\left(e_{l-1,n}^{(m)+},\vartheta_{l-1,n}^{(k)}\right)\\&\quad+\int_{I_{l,n}}\left(-\dot e^{(m)}+\overline{F'(z^{(m)})}e^{(m)},\vartheta^{(k)}\right)dt+\int_{I_{l,n}}\left(e^{(m)},\eta^{(k)}\right)dt.\end{aligned} \tag{26}$$

We note that

$$\begin{aligned}-\dot e^{(m)}+\overline{F'(z^{(m)})}e^{(m)}&=\left(-\dot y^{(m)}+F\left(y^{(m)}\right)\right)+\left(\dot Y^{(m)}-F\left(Y^{(m)}\right)\right)\\&=-\delta_y^{(m)}+\dot Y^{(m)}-F\left(Y^{(m)}\right).\end{aligned}$$

Furthermore, using continuity of $y^{(m)}$,

$$\begin{aligned}e_{l-1,n}^{(m)+}&=y_{l-1,n}^{(m)+}-Y_{l-1,n}^{(m)+}=\left(y_{l-1,n}^{(m)-}-Y_{l-1,n}^{(m)-}\right)-\left(Y_{l-1,n}^{(m)+}-Y_{l-1,n}^{(m)-}\right)\\&=e_{l-1,n}^{(m)-}-\left[Y^{(m)}\right]_{l-1,n}.\end{aligned}$$

Inserting these expressions into (26) yields the recursive relation

$$\begin{aligned}\left(e_{l,n}^{(m)-},\vartheta_{l,n}^{(k)}\right)&=\left(e_{l-1,n}^{(m)-},\vartheta_{l-1,n}^{(k)}\right)-\int_{I_{l,n}}\left(\dot Y^{(m)}-F(Y^{(m)}),\vartheta^{(k)}\right)dt\\&\quad-\left([Y^{(m)}]_{l-1,n},\vartheta_{l-1,n}^{(k)}\right)+\int_{I_{l,n}}\left(\delta_y^{(m)},\vartheta^{(k)}\right)dt\\&\quad-\int_{I_{l,n}}\left(e^{(m)},\eta^{(k)}\right)dt.\end{aligned} \tag{27}$$

This is the basis for the equation for the error at time t_n stated in the following lemma.

Lemma 4.2. *The iterative multirate finite element methad satisfies an error equation aver one time step:*

$$(e_n^{(m)-},\psi_n)=\left(e_{n-1}^{(m)-},\vartheta_{n-1}^{(k)}\right)+Q_{1,n}+Q_{2,n}+\sum_{l=1}^{L_{1,n}}\int_{I_{l,n}}\left(\delta_y^{(m)}+\delta_Y^{(m)},\vartheta^{(k)}\right)dt-\sum_{l=1}^{L_{1,n}}\int_{I_{l,n}}\left(e^{(m)},\eta^{(k)}\right)dt,$$

where

$$Q_{1,n}=-\sum_{l=1}^{L_{1,n}}\left\{\int_{I_{l,n}}\left(\dot Y_1^{(m)}-F_1\left(Y_1^{(m)},Y_2^{(m-1)}\right),\vartheta_1^{(k)}\right)dt+\left(\left[Y_1^{(m)}\right]_{l-1,n},\vartheta_{1,l-1,n}^{(k)}\right)\right\},$$

$$Q_{2,n}=-\sum_{l=1}^{L_{1,n}}\left\{\int_{I_{l,n}}\left(\dot Y_2^{(m)}-F_2\left(Y_1^{(m)},Y_2^{(m)}\right),\vartheta_2^{(k)}\right)dt+\left(\left[Y_2^{(m)}\right]_{l-1,n},\vartheta_{2,l-1,n}^{(k)}\right)\right\},$$

and $\delta_Y^{(m)}$ *is defined analogausly ta* $\delta_y^{(m)}$.

Proof. This is obtained by using the recursive relation (27). □

We note that this equation reflects the error arising from the consistent finite element numerical discretization of the analytical fixed point iteration. Similar to Lemma 4.1, this error contains the iteration residuals weighted by the adjoint $\vartheta^{(k)}$. The last term is not computable since it contains the error $e^{(m)}$ weighted by the iteration residual in the adjoint computation. Again provided that an *a priori* estimate on $e^{(m)}$ is available, we can bound this term as higher order due the fact that the residual can be made as small as needed when the adjoint computation is driven to convergence.

We now collect all the resulting estimates and obtain an error estimate of the finite element multiscale iterative implicit method by setting $y - Y^{(m)} = (y - y^{(m)}) + (y^{(m)} - Y^{(m)})$.

Theorem 4.2. *Set $\psi_N = \psi$ and $\psi_{n-1} = \vartheta_{n-1}^{(K_n)}$ for $n = N, N-1, \ldots, 2$. Then the error of iterative multirate Galerkin finite element method at final time $t_N = T$ can be expressed as*

$$\left(y_N - Y_N^{(M_N)-}, \psi\right) = \sum_{n=1}^{N}(Q_{1,n} + Q_{2,n} + Q_{3,n} + Q_{4,n} + Q_{5,n} + Q_{6,n}), \tag{28}$$

$Q_{1,n}$ and $Q_{2,n}$ are given in Lemma 4.2 with m replaced by M_n and k replaced by K_n, and

$$Q_{3,n} = \sum_{l=1}^{L_{1,n}} \int_{I_{l,n}} \left(\delta_Y^{(M_n)}, \vartheta^{(K_n)}\right) dt$$

$$Q_{4,n} = \sum_{l=1}^{L_{1,n}} \int_{I_{l,n}} \left(\delta_y^{(M_n)}, \vartheta^{(K_n)} - \varphi^{(K_n)}\right) dt$$

$$Q_{5,n} = \left(y_{n-1}^{(M_n)}, \Delta\Phi_n^{(K_n)}\psi_n\right) + \int_{I_n} \left(y^{(M_n)}, \xi^{(K_n)}\right) dt$$

$$Q_{6,n} = \left(y_{n-1} - y_{n-1}^{(M_n)}, \Delta\Phi_n^{(K_n)}\psi_n\right) - \sum_{l=1}^{L_{1,n}} \int_{I_{l,n}} \left(e^{(M_n)}, \eta^{(K_n)}\right) dt,$$

Proof. Denote $e^{(m)} = y - Y^{(m)}$. First we need to get the total error over one time step. We set $y_{n-1}^{(m)} = y_{n-1}$ in Lemma 4.2 and combine it with Lemma 4.1 to get

$$\begin{aligned}\left(e_n^{(M_n)-}, \psi_n\right) &= \left(e_{n-1}^{(M_n)-}, \vartheta_{n-1}^{(K_n)}\right) + Q_{1,n} + Q_{2,n} + Q_{3,n} + Q_{4,n} \\ &\quad + \left(y_{n-1}, \Delta\Phi_n^{(K_n)}\psi_n\right) + \int_{I_n} \left(y^{(M_n)}, \xi^{(K_n)}\right) dt \\ &\quad - \sum_{l=1}^{L_{1,n}} \int_{I_{l,n}} \left(e^{(M_n)}, \eta^{(K_n)}\right) dt. \end{aligned} \tag{29}$$

We note that since $Y_{n-1}^{(M_n)-} = Y_{n-1}^{(M_{n-1})-}$ (see Algorithm 1), we have $e_{n-1}^{(M_n)-} = e_{n-1}^{(M_{n-1})-}$. Furthermore, by adding and subtracting $\left(y_{n-1}^{(M_n)}, \Delta\Phi_n^{(K_n)}\psi_n\right)$, we get

$$\left(y_{n-1}, \Delta\Phi_n^{(K_n)}\psi_n\right) + \int_{I_n} \left(y^{(M_n)}, \xi^{(K_n)}\right) dt - \sum_{l=1}^{L_{1,n}} \int_{I_{l,n}} \left(e^{(M_n)}, \eta^{(K_n)}\right) dt = Q_{5,n} + Q_{6,n}.$$

Combining all this expressions in (29) yield a recursive relation for the total error over one time step. The error at the final time is obtained from undoing this relation. □

Theorem 4.2 has decomposed the total error at the final time into several components. The term $Q_{1,n}$ represents the finite element residual associated with the fast time scale subsystem, while $Q_{2,n}$ represents the finite element residual associated with the slow time scale. The term $Q_{3,n}$ represents the iteration error quantified by the iteration residual $\delta_Y^{(M_n)}$. It is expected that once convergence is reached this component should vanish. The term $Q_{4,n}$ also contains the iteration residual, so when convergence is reached, this component vanishes as well. Moreover, we note that in this term, the iteration residual is weighted by $\vartheta^{(K_n)} - \varphi^{(K_n)}$. Recall that the adjoints ϑ and φ differ in the functions which are used for linearization. Thus, the term $Q_{4,n}$ also vanishes when $\vartheta^{(k)} = \varphi^{(k)}$, which may be true if, for example, the system (1) is linearly coupled, i.e. if $F_i(y_1, y_2) = A_{i1}y_1 + A_{i2}y_2$ for some matrix A_{i1} and A_{i2}.

The term $Q_{5,n}$ and $Q_{6,n}$ contains $\Delta\Phi_n^{(k)}\psi_n$ which is not computable. As has been mentioned, provided an *a priori* estimate regarding the error of y and Y is available, $Q_{6,n}$ is of higher order in the asymptotic limit. All these issues are addressed in the next section.

4.2.3. A computable error estimate

The following lemma shows that if the analytic fixed point iteration has small residual, then $\Delta\Phi_n^{(k)}\psi_n$ can be written as a sum of the residuals of the adjoint iterations and some higher order terms.

Lemma 4.3. *If F is Lipschitz continuous and $y^{(M_n)}$ is sufficiently close to y in I_n,*

$$\Delta\Phi_n^{(K_n)}\psi_n = -\int_{I_n} \xi^{(K_n)}\, dt + \text{h.o.t.}$$

Proof. We denote by $\hat{\varphi}$ function that satisfy

$$\begin{cases} -\dot{\hat{\varphi}} = \overline{F'(y^{(M_n)})}^{\top}\hat{\varphi}, & t \in (t_n, t_{n-1}), \\ \hat{\varphi}(t_n) = \psi_n. \end{cases} \tag{30}$$

Using this equation we write $\varphi - \varphi^{(K_n)} = (\varphi - \hat{\varphi}) + (\hat{\varphi} - \varphi^{(K_n)}) = A + B$, where A satisfies

$$\begin{cases} -\dot{A} = \overline{F'(y^{(M_n)})}A + \left[\overline{F'(y)} - \overline{F'(y^{(M_n)})}\right]\varphi, & t \in (t_n, t_{n-1}) \\ A(t_n) = 0. \end{cases}$$

Using a standard technique for system of ordinary differential equations, we get

$$A(t) = \int_t^{t_n} \left[\overline{F'(y)} - \overline{F'(y^{(M_n)})}\right]\varphi\, d\tau + \mathbf{O}\left(\Delta t_n^2 \left|\overline{F'(y)} - \overline{F'(y^{(M_n)})}\right|\right).$$

If F is Lipschitz continuous,

$$|A(t)| \leqslant C\Delta t_n |y - y^{(M_n)}||\varphi|,$$

and thus, when $y^{(M_n)}$ is sufficiently close to y in I_n, $A(t)$ is of higher order. Using (21), B satisfies

$$\begin{cases} -\dot{B} = \overline{F'(y^{(M_n)})}B - \xi^{(K_n)} & t \in (t_n, t_{n-1}), \\ B(t_n) = 0, \end{cases}$$

with solution expressed in similar fashion as $A(t)$,

$$B(t) = -\int_t^{t_n} \xi^{(K_n)}\, d\tau + \mathbf{O}\left(\xi^{(K_n)}\Delta t_n^2\right).$$

Computing $B(t_{n-1})$ completes the proof. □

Once this is in place, we may verify that $\sum_{n=1}^{N} Q_{5,n}$ and $\sum_{n=1}^{N} Q_{6,n}$ are of higher order. These are stated in the following lemma.

Lemma 4.4. *When $\xi^{(K_n)}$ and $\eta^{(K_n)}$ are sufficiently small, the terms $\sum_{n=1}^{N} Q_{5,n}$ and $\sum_{n=1}^{N} Q_{6,n}$ are of higher order.*

Proof. Using Lemma 4.3 we get

$$Q_{5,n} = \int_{I_n} \left(y^{(M_n)} - y_{n-1}^{(M_n)}, \xi^{(K_n)}\right) dt + \text{h.o.t.}$$

Since $|y^{(M_n)} - y_{n-1}^{(M_n)}| \leqslant C\Delta t_n$, $Q_{5,n} \approx \mathbf{O}(\xi^{(K_n)}\Delta t_n^2)$ which for sufficiently small $\xi^{(K_n)}$ makes $\sum_{n=1}^{N} Q_{5,n}$ a higher order term. Similarly, for sufficiently small $\xi^{(K_n)}$ and $\eta^{(K_n)}$, $\sum_{n=1}^{N} Q_{6,n}$ is also a higher order term.

because both these residuals are weighted by the numerical solution errors. □

Based on Theorem 4.2 and incorporating the two lemmas above, we may now write a computable error estimator for the iterative multirate Galerkin finite element method.

Theorem 4.3. *The computable error of iterative multirate Galerkin finite element method at final time $t_N = T$ is*

$$\left(y_N - Y_N^{(M_N)-}, \psi\right) \approx Q_1 + Q_2 + Q_3 + Q_4 = \sum_{n=1}^{N} (Q_{1,n} + Q_{2,n} + Q_{3,n} + Q_{4,n}), \tag{31}$$

where

$$Q_{1,n} = -\sum_{l=1}^{L_{1,n}} \left\{ \int_{I_{l,n}} \left(\dot{Y}_1^{(M_n)} - F_1\left(Y_1^{(M_n)}, Y_2^{(M_n-1)}\right), \vartheta_1^{(K_n)} \right) dt + \left(\left[Y_1^{(M_n)} \right]_{l-1,n}, \vartheta_{1,l-1,n}^{(K_n)} \right) \right\},$$

$$Q_{2,n} = -\sum_{l=1}^{L_{1,n}} \left\{ \int_{I_{l,n}} \left(\dot{Y}_2^{(M_n)} - F_2\left(Y_1^{(M_n)}, Y_2^{(M_n)}\right), \vartheta_2^{(K_n)} \right) dt + \left(\left[Y_2^{(M_n)} \right]_{l-1,n}, \vartheta_{2,l-1,n}^{(K_n)} \right) \right\}$$

$$Q_{3,n} = \sum_{l=1}^{L_{1,n}} \int_{I_{l,n}} \left(\delta_Y^{(M_n)}, \vartheta^{(K_n)} \right) dt$$

$$Q_{4,n} = \sum_{l=1}^{L_{1,n}} \int_{I_{l,n}} \left(\delta_Y^{(M_n)}, \vartheta^{(K_n)} - \varphi^{(K_n)} \right) dt.$$

Remark 4.1. Notice that $Q_{4,n}$ contains $\delta_y^{(M_n)}$ which is an expression that is dependent on $y^{(M_n)}$, the analytic fixed point iteration solution of (1). By adding and subtracting $\delta_Y^{(M_n)}$ to $Q_{4,n}$ we get

$$Q_{4,n} = \sum_{l=1}^{L_{1,n}} \int_{I_{l,n}} \left(\delta_Y^{(M_n)}, \vartheta^{(K_n)} - \varphi^{(K_n)} \right) dt + \sum_{l=1}^{L_{1,n}} \int_{I_{l,n}} \left(\delta_y^{(M_n)} - \delta_Y^{(M_n)}, \vartheta^{(K_n)} - \varphi^{(K_n)} \right) dt. \tag{32}$$

The second term in the last equation is higher order because it involves the difference between two residuals.

4.3. A computable error estimate including projection

Relaxing our assumption in Section 4.2 and allowing for a projection other than the identity, we may write a computable error estimate for the iterative multirate Galerkin finite element method including projection.

Theorem 4.4. *The computable error of iterative multirate Galerkin finite element method including projection at final time $t_N = T$ is*

$$\left(y_N - Y_N^{(M_N)-}, \psi\right) \approx Q_1 + Q_2 + Q_3 + Q_4 + Q_{\Pi} = \sum_{n=1}^{N} (Q_{1,n} + Q_{2,n} + Q_{3,n} + Q_{4,n} + Q_{\Pi,n}), \tag{33}$$

where

$$Q_{1,n} = -\sum_{l=1}^{L_{1,n}} \left\{ \int_{I_{l,n}} \left(\dot{Y}_1^{(M_n)} - F_1\left(Y_1^{(M_n)}, Y_2^{(M_n-1)}\right), \vartheta_1^{(K_n)} \right) dt + \left(\left[Y_1^{(M_n)} \right]_{l-1,n}, \vartheta_{1,l-1,n}^{(K_n)} \right) \right\},$$

$$Q_{2,n} = -\sum_{l=1}^{L_{1,n}} \left\{ \int_{I_{l,n}} \left(\dot{Y}_2^{(M_n)} - F_2\left(\Pi Y_1^{(M_n)}, Y_2^{(M_n)}\right), \vartheta_2^{(K_n)} \right) dt + \left(\left[Y_2^{(M_n)} \right]_{l-1,n}, \vartheta_{2,l-1,n}^{(K_n)} \right) \right\}$$

$$Q_{3,n} = \sum_{l=1}^{L_{1,n}} \int_{I_{l,n}} \left(\delta_Y^{(M_n)}, \vartheta^{(K_n)} \right) dt$$

$$Q_{4,n} = \sum_{l=1}^{L_{1,n}} \int_{I_{l,n}} \left(\delta_Y^{(M_n)}, \vartheta^{(K_n)} - \varphi^{(K_n)} \right) dt$$

$$Q_{\Pi,n} = \sum_{k=1}^{L_{2,n}} \int_{I_{k,n}} \left(F_2\left(Y_1^{(M_n)}, Y_2^{(M_n)}\right) - F_2\left(\Pi Y_1^{(M_n)}, Y_2^{(M_n)}\right), \vartheta_2^{(K_n)} \right) dt.$$

5. Numerical experiments

In this section, we present several numerical examples that show the performance of the error estimates. All forward problems are solved using the lowest order, piecewise constant dG method, which is equivalent to backward Euler scheme. In particular, for iteration level m, the scheme is

$$\begin{cases} Y_{1,j,n}^{(m)} - Y_{1,j-1,n}^{(m)} = \Delta s_1 F_1\left(Y_{1,j,n}^{(m)}, Y_{2,k,n}^{(m-1)}\right), \quad j = 1, 2, \ldots, L_{1,n} \\ Y_{2,k,n}^{(m)} - Y_{2,k-1,n}^{(m)} = \Delta s_1 \sum_{j=1}^{d_n} F_2\left(\Pi Y_{1,l_j,n}^{(m)}, Y_{2,k,n}^{(m)}\right), \\ k = 1, 2, \ldots, L_{2,n}, \quad l_j = (k-1)d_n + j. \end{cases}$$

When nonlinear, the individual component equations are solved using Newton's Method. The adjoint solutions are computed using a second order, piecewise linear, continuous Galerkin method, which is equivalent to the second order Crank–Nicolson scheme.

In order to illuminate the behavior of the error, we take the quantity of interest to be the individual error in each component at the final time. We point out that the choice of the quantity of interest has a significant impact on the behavior of the error in general [26,23]. This is even more significant in a multiscale problem.

We demonstrate the robustness of the proposed error estimator through several examples below. These examples also show the potential for using an accurate estimate to adaptively determine the parameters controlling accuracy. Since the error estimate is written as a sum of contributing components, we can determine the largest source of error and adjust the corresponding parameter.

In the first example in Section 5.1, we illustrate the consequences of projections between scales. The rest of the examples illustrate the consequences of incomplete iterations, and in those examples we assume Π = the identity.

5.1. Numerical exomple illustroting discretization ond projection error

To illustrate the performance of the error estimator provided by Theorem 4.1 (fully implicit multirate Galerkin finite element method), we consider the numerical solution of a 3 × 3 system

$$\begin{cases} \dot{x} = 100y + z, \quad x(0) = \frac{9001}{10001} \\ \dot{y} = -100x, \quad y(0) = \frac{10^5}{10001} \\ \dot{z} = -\frac{z}{10001}((10001x + z)^2 + (10001y + 100z)^2), \quad z(0) = 1000, \end{cases} \tag{34}$$

which has fast and slow equations coupled nonlinearly. In particular, the equation determining $z(t)$ contains nonlinear coupling to the fast scale components $x(t)$ and $y(t)$. The true solution is $x(t) = \cos(100t) - \frac{1000}{10001}e^{-t}$, $y(t) = -\sin(100t) - \frac{10^5}{10001}e^{-t}$, and $z(t) = 1000e^{-t}$. There are two distinct time scales, fast $\mathbf{O}(2\pi/100)$ and slow $\mathbf{O}(1)$. We set $y_1 = [x\ y]^\top$ (associated with the fast time scale) and $y_2 = z$ (associated with the slow time scale). A typical solution is depicted in Fig. 3.

The multirate finite element solution is constructed on the piecewise constant finite element space, i.e., with $q_1 = q_2 = 0$. The system is solved until $T = 0.5$. We use $\Delta t = 0.5/10$, $\Delta s_1 = \Delta t/800$, and $\Delta s_2 = \Delta t/10$ and examine three different projections; (i) Π_1: the identity operator, (ii) Π_2: averaging over $(t_{k-1,n}, t_{k,n})$ (i.e., over a subinterval of length Δs_2), and (iii) Π_3: averaging over (t_{n-1}, t_n). Fig. 4 compares the exact errors of the multirate solutions when solved employing these three different projections. As expected, the multirate solutions exhibit the best performance when the identity operator is used. In all subsequent examples we shall assume that the projection is the identity operator and thus $Q_{\Pi} = 0$.

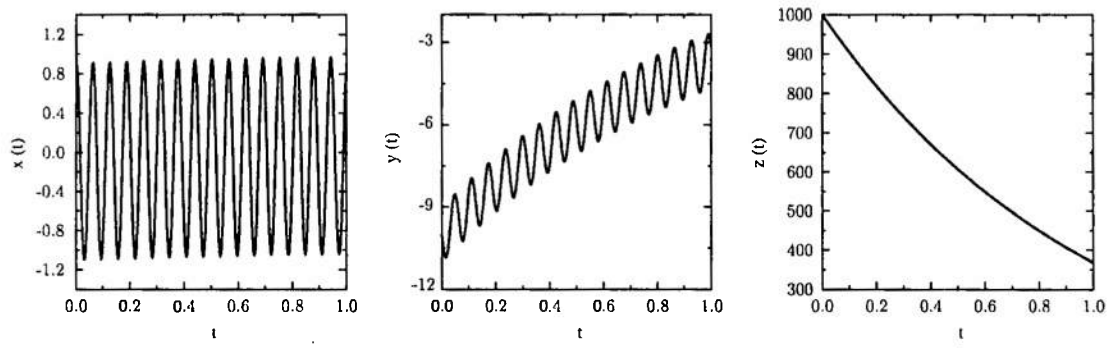


Fig. 3. Typical solution of (34).

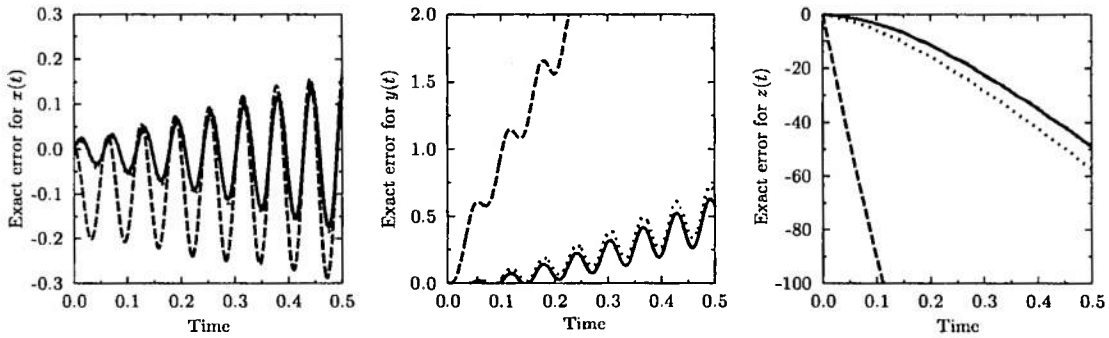


Fig. 4. Comparison of exact errors for multirate solution of (34) with various projection. Π_1: identity (solid lines), Π_2: local average over $I_{k,n}$ (dotted lines), and Π_3: average over I_n (dashed lines).

Table 1
Performance of error estimator at $T = 0.5$ for Π_1, the identity operator.

	Error for $x(t)$	Error for $y(t)$	Error for $z(t)$
Exact error	0.124	0.569	−48.92
Error estimate	0.126	0.535	−46.11
$\mathcal{Q}_1$	0.123	0.500	−43.63
$\mathcal{Q}_2$	0.003	0.035	−2.48
$\mathcal{Q}_\Pi$	0	0	0

Table 2
Performance of error estimator at $T = 0.5$ for Π_2, averaging over $I_{k,n}$.

	Error for $x(t)$	Error for $y(t)$	Error for $z(t)$
Exact error	0.126	0.665	−57.31
Error estimate	0.128	0.541	−46.81
$\mathcal{Q}_1$	0.123	0.499	−43.66
$\mathcal{Q}_2$	−0.014	−0.144	10.79
$\mathcal{Q}_\Pi$	0.019	0.186	−13.94

Table 3
Performance of error estimator at $T = 0.5$ for Π_3, averaging over I_n.

	Error for $x(t)$	Error for $y(t)$	Error for $z(t)$
Exact error	0.161	3.775	−370.03
Error estimate	0.194	0.970	−87.64
$\mathcal{Q}_1$	0.111	0.540	−48.63
$\mathcal{Q}_2$	0.004	0.018	−1.64
$\mathcal{Q}_\Pi$	0.079	0.412	−37.37

Tables 1–3 show the performance of error estimator when solving the system employing the three different projections. As expected, the estimator performs reasonably well when Π_1 and Π_2, the identity and local averaging operators respectively, are used and it breaks down when Π_3, the averaging operator over I_n, is used. Nevertheless, the estimator still gives a hint of what is actually happening in terms of the relative size of the projection error $\mathcal{Q}_\Pi$ compared to the total estimated error for all three components.

5.2. A one-way system with the fast variables coupled into the slow equation

We consider the 3 × 3 system

$$\begin{cases} \dot{x} = -50y, & x(0) = 1 \\ \dot{y} = 50x, & y(0) = 0 \\ \dot{z} = -z + x + y, & z(0) = 2, \end{cases} \tag{35}$$

in which the fast variables are coupled into the equation for the slow subsystem. The true solution is $x(t) = \cos(50t)$, $y(t) = \sin(50t)$, and $z(t) = \frac{5051}{2501}e^{-t} - \frac{49}{2501}\cos(50t) + \frac{51}{2501}\sin(50t)$. We set $y_1 = [x\ y]^T$ (associated with the fast time scale) and $y_2 = z$ (associated with the slow time scale). Since the coupling is one way, there is no iteration needed when solving the system, i.e. we solve for y_1 and use the solution to solve for y_2. The same holds for the associated adjoint computation. Thus, the error arises solely from the numerical solution of the fast and slow subsystems. Note however that the numerical error of the fast component affects the accuracy of the slow component. We plot the typical behavior of the error in Fig. 5. The accuracy of the method deteriorates for longer times, however the estimator can accurately predict the error dynamics.

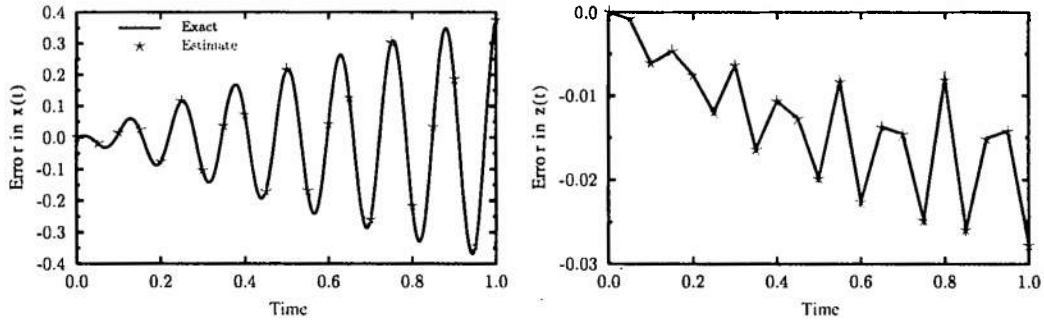


Fig. 5. Time history of the error for solving (35). Left: $x(t)$. Right: $z(t)$. The time steps $\Delta t = 0.05$, $\Delta s_1 = \Delta t/128$, $\Delta s_2 = \Delta t$.

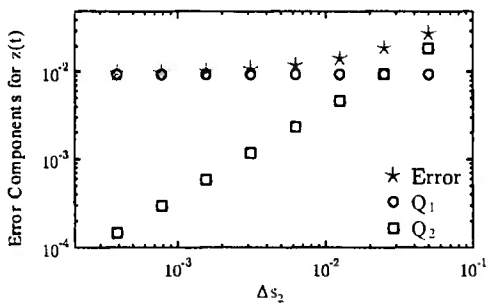


Fig. 6. Error for component $z(t)$ in (35) at $T = 1$ plotted against Δs_2.

Only the terms Q_1 and Q_2 contribute to the error estimate (31), and in fact, the error is dominated by the finite element residual from the fast scale Q_1. Fig. 6 shows the error in component $z(t)$ at the final time $T = 1$ when the system is solved using $\Delta t = 0.05$, $\Delta s_1 = \Delta t/128$, and decreasing Δs_2. The slow scale finite element residual Q_2 decreases linearly as Δs_2 decreases. On the other hand, the fast scale finite element residual Q_1 does not exhibit significant change. Apparently, decreasing Δs_2 yields improved accuracy only until a certain stage, after which the error is dominated by the fast scale residual. In terms of adaptivity, this example emphasizes the potential of the error estimator to provide criteria for time step refinement specific to the dominant error component.

5.3. A system with a slow variable coupled into the fast equations

Next, we consider the 3 × 3 system

$$\begin{cases} \dot{x} = 100y + z, & x(0) = \frac{9001}{10001} \\ \dot{y} = -100x, & y(0) = \frac{10^5}{10001} \\ \dot{z} = -z, & z(0) = 1000, \end{cases} \tag{36}$$

in which the slow variable enters into the fast equations. In fact, this system has the same solution as (34) in Section 5.1. As in that subsection, we set $y_1 = [x\ y]^T$ and $y_2 = z$. Because the slow scale equation does not involve the fast scale variables, two iterations are sufficient to reduce the iteration error. Also, the slow scale finite element residual component of the error Q_1 is zero for component y_2. For all components, the iteration residual component Q_4 is zero because the adjoints φ and ϑ are equal, which is a consequence of the fact that $\frac{\partial F_i}{\partial y_j}$ is a constant independent of the solution.

Fig. 7 shows the error components in the fast scale components $x(t)$ and $y(t)$ as a function of the fast time step Δs_1. Here (36) is solved until $T = 2$ with $\Delta t = 0.2$ and $\Delta s_2 = \Delta t/20$. The method uses only one iteration in each of the coarse time steps Δt. The slow scale component $z(t)$ has been solved accurately with error about 0.5%. Moreover, the difference between estimated and exact errors is about 0.08%. As shown in the figure, the error estimator gives an accurate prediction despite the inaccuracy of the method. Each component exhibits a different error behavior in terms of the dominant component. For component $x(t)$, the dominant component is Q_3, i.e. the iteration error. Obviously decreasing Δs_1 does not help improving the method's accuracy. The fast scale finite element residual Q_1 does seem to decrease linearly with respect to Δs_1. By contrast, the error in component $y(t)$ is dominated by Q_1, and thus decreasing Δs_1 would result in smaller Q_1 and hence reducing the error for this component. Moreover, when Δs_1 is sufficiently small, the contribution of error from all components becomes relatively comparable.

Fig. 8 shows the error components in the fast scale components $x(t)$ and $y(t)$ when two iterations are used to solve the system. In this case, the iteration error component Q_3 is essentially zero and the dominant component is Q_1 for both $x(t)$ and $y(t)$. As Δs_1 is decreased, this term decreases as does the total error. Both components exhibit similar behavior and the error estimator predicts the exact error accurately.

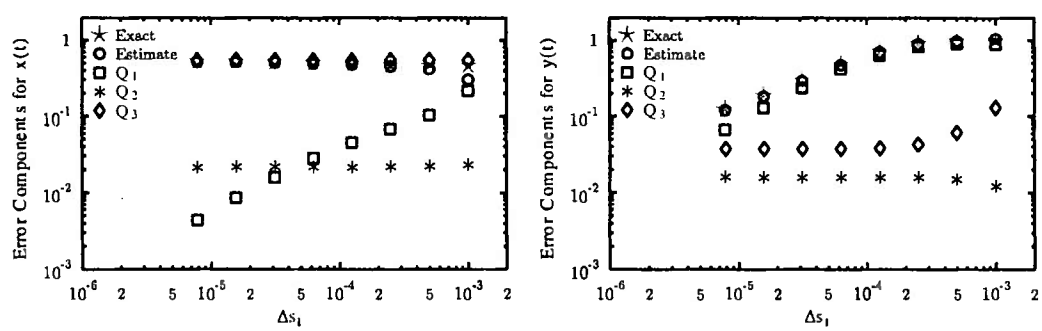


Fig. 7. Error for components $x(t)$ and $y(t)$ in (36) at $T = 2$. Solutions are obtained using one iteration.

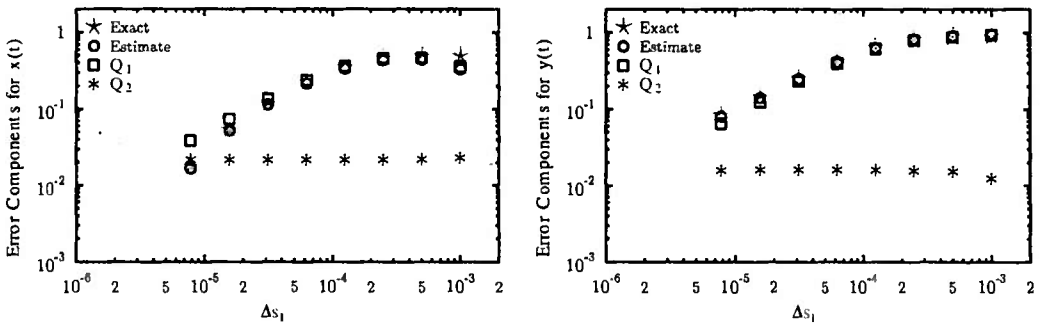


Fig. 8. Error for components $x(t)$ and $y(t)$ in (36) at $T = 2$. Solutions are obtained using two iterations.

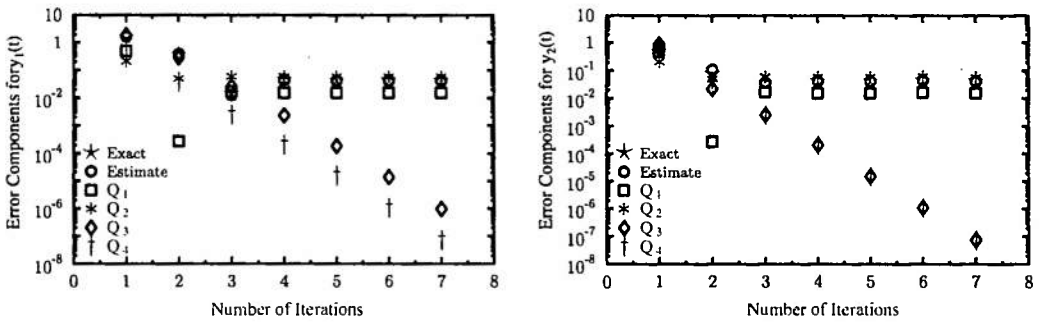


Fig. 9. Error for components $y_1(t)$ and $y_2(t)$ at $T = 1$ in (37) as a function of number of iterations.

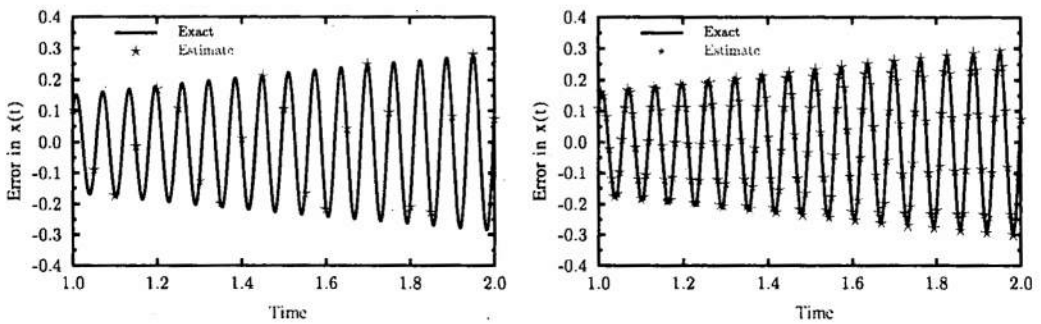


Fig. 10. Time history of error in x for solving (34). Left: $\Delta t = 0.05, \Delta s_1 = \Delta t/1600$, $\Delta s_2 = \Delta t/32$. Right: $\Delta t = 0.00625$, $\Delta s_1 = \Delta t/200$, $\Delta s_2 = \Delta t/4$.

5.4. A nonlinearly coupled system with one scale

We consider

$$\begin{cases} \dot{x} = e^{y_1} + e^{y_2} - 2, & y_1(0) = -1, \\ \dot{y} = -e^{y_1} - e^{y_2} + 2, & y_2(0) = 1. \end{cases} \tag{37}$$

The exact solution is $y_1(t) = \ln((e-1)t+1) - \ln((e-1)t+e)$, and $y_2(t) = -y_1(t)$. This system is not multiscaled, however the two equations are coupled in nonlinear fashion and we can investigate the behavior of the error as the iterations increase. Fig. 9 shows the behavior of the error over $[0,1]$ with $\Delta t = 1$, and $\Delta s_1 = \Delta s_2 = \Delta t/4$. At the first iteration, all error components are relatively comparable to each other. As the iterations increase, the components Q_3 and Q_4 are reduced. However, the overall error fails to continue to improve significantly as the iterations increase because the error is eventually dominated by the finite element residuals Q_1 and Q_2. We can see also that in each iteration the error estimator is in good agreement with the exact error.

5.5. A nonlinearly coupled multiscale system

Next, we reconsider the 3×3 system in (34) described in Section 5.1. Figs. 10–12 shows the time history of the error for the three components. The system is solved until $T = 2$. The plots on the left are for $\Delta t = 0.05$ and on the right for $\Delta t = 0.00625$. We maintain the absolute size of the other time steps, resulting in $\Delta s_1 = 0.05/1600$ for the left column plots, and $\Delta s_1 = 0.00625/200$ for the right columns plots. The error in all plots are shown for the solutions obtained after convergence is reached. It requires 5 iterations to reach convergence for the solution with $\Delta t = 0.05$ (lefthand plots), and only 2 iterations for the solution with

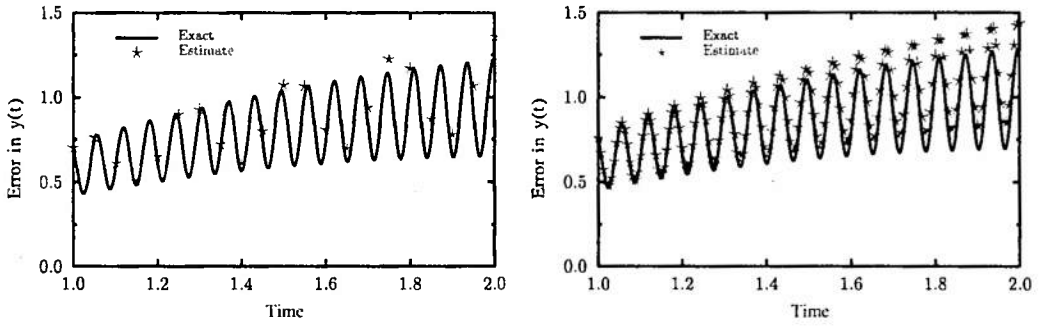


Fig. 11. Time history of error in y for solving (34). Left: $\Delta t = 0.05$, $\Delta s_1 = \Delta t/1600$, $\Delta s_2 = \Delta t/32$. Right: $\Delta t = 0.00625$, $\Delta s_1 = \Delta t/200$, $\Delta s_2 = \Delta t/4$.

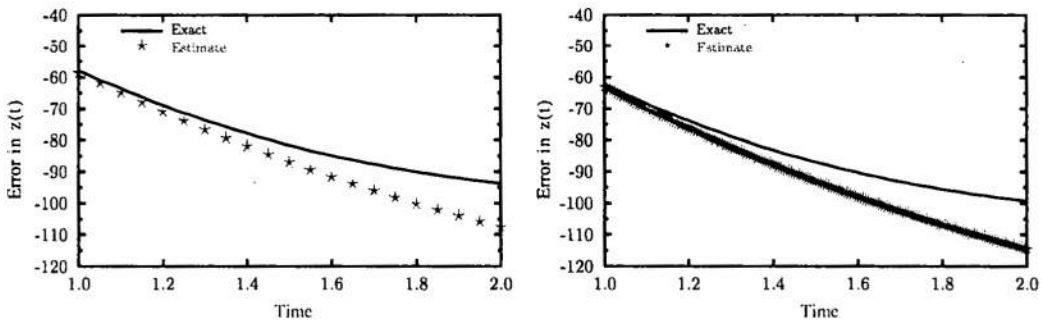


Fig. 12. Time history of error in z for solving (34). Left: $\Delta t = 0.05$, $\Delta s_1 = \Delta t/1600$, $\Delta s_2 = \Delta t/32$. Right: $\Delta t = 0.00625$, $\Delta s_1 = \Delta t/200$, $\Delta s_2 = \Delta t/4$.

$\Delta t = 0.00625$ (righthand plots). The dominant error is always the fast scale finite element residual Q_1. The estimator predicts the error with ≈2% difference for $x(t)$, and ≈11% for $y(t)$ and $z(t)$.

5.6. A two-scale system of wire suspended masses

We return to the two-scale version of the mass-wire system (2) described in the introduction. We set $M = 10$, $m = 0.1$, $A = 0.25$, $a = 0.1$, and $\Gamma = \gamma = 0$.

Fig. 13–16 show the time history of the error of three components of the solution, the slow component/heavy mass at location x_1, the light mass at x_3 (the so-called "bridge" mass) that is connected to a heavy mass on one side and a light mass on the other, and the fast component/light mass at location x_7. Even when the error is very large, the error estimate gives an accurate picture of the error. The figures also indicate the dominant component in each case. For example, when only one iteration is used, the iteration error is dominant, while when three iterations is used, the finite element residual is the dominating component.

6. Details of the *a priori* analysis

6.1. Convergence of the analytic fixed point iteration

As with the standard local analysis for ordinary differential equations, the solution of (9) is sought in a neighborhood of the initial condition $y(t_{n-1})$. Because $F \in C^1(E)$, it is locally Lipschitz in E. In particular, for $y(t_{n-1}) \in E$, there exists an ϵ neighborhood $\mathcal{B}_\epsilon(y(t_{n-1}))$ in E and a positive constant $\mathcal{L}$ such that $|F(u) - F(v)| \leqslant \mathcal{L}|u - v|$, for all u and v in $\mathcal{B}_\epsilon(y(t_{n-1}))$. In addition, with $b = \epsilon/2$, the function F is continuous and bounded with bound $\mathcal{M}$ in the compact set $\overline{\mathcal{B}} = \{u \in \mathbb{R}^2,$ such that $|u - y(t_{n-1})| \leqslant b\}$. We claim that the solution of (9) is unique in $\overline{\mathcal{B}}$. It is obvious that the argument employed to achieve this is exactly the same for each integral equation because for fixed m, each integral equation is solved independently of each other. The following lemma can be applied appropriately to each of the integral equations in (9).

Lemma 6.1. *Assume that $(\alpha, \beta) \in \overline{\mathcal{B}}$. Then the integral equation*

$$\xi(t) = \alpha + \int_{t_{n-1}}^{t} F_i(\xi, \beta)\, ds, \tag{38}$$

admits a unique solution with $(\xi, \beta) \in \overline{\mathcal{B}}$.

Proof. Part of the proof closely follows standard arguments for existence and uniqueness (see [47]). We set $\xi^{(0)} = \alpha$ and compute

$$\xi^{(j)}(t) = \alpha + \int_{t_{n-1}}^{t} F_i\left(\xi^{(j-1)}, \beta\right) ds, \tag{39}$$

for $j = 1, 2, \ldots$. For $j = 1$, this gives $|\xi^{(1)}(t) - \alpha| \leqslant \mathcal{M}(t - t_{n-1}) \leqslant \mathcal{M}\Delta t_n$. Then by choosing $\Delta t_n \leqslant b/\mathcal{M}$ (and thus $t_n \leqslant t_{n-1} + b/\mathcal{M}$) we have $(\xi^{(1)}, \beta) \in \overline{\mathcal{B}}$. By induction, $(\xi^{(j)}, \beta) \in \overline{\mathcal{B}}$. We proceed to show that the sequence $\{\xi^{(j)}\}$ converges an element of $\overline{\mathcal{B}}$. Using the Lipschitz condition, $|\xi^{(1)}(t) - \xi^{(0)}| \leqslant (t - t_{n-1})\mathcal{M}$, and induction gives

$$|\xi^{(j)}(t) - \xi^{(j-1)}(t)| \leqslant \frac{\mathcal{M}}{\mathcal{L}} \frac{(\mathcal{L}\Delta t_n)^j}{j!} \leqslant \frac{\mathcal{M}}{\mathcal{L}} (\mathcal{L}\Delta t_n)^j,$$

for $j \geqslant 2$. As long as $\Delta t_n \mathcal{L} < 1$, we know that for $l > k > N$

$$|\xi^{(l)}(t) - \xi^{(k)}(t)| \leqslant \sum_{j=k+1}^{l} |\xi^{(j)}(t) - \xi^{(j-1)}(t)| \leqslant \frac{\mathcal{M}}{\mathcal{L}} \frac{(\mathcal{L}\Delta t_n)^N}{1 - \mathcal{L}\Delta t_n}.$$

By choosing $\Delta t_n < \min\{b/\mathcal{M}, 1/\mathcal{L}\}$, $|\xi^{(l)}(t) - \xi^{(k)}(t)|$ vanishes as $N \to \infty$. This implies that $\xi^{(j)}(t)$ is a Cauchy sequence of continuous functions in $I_n = [t_{n-1}, t_n]$ which converges uniformly to an element

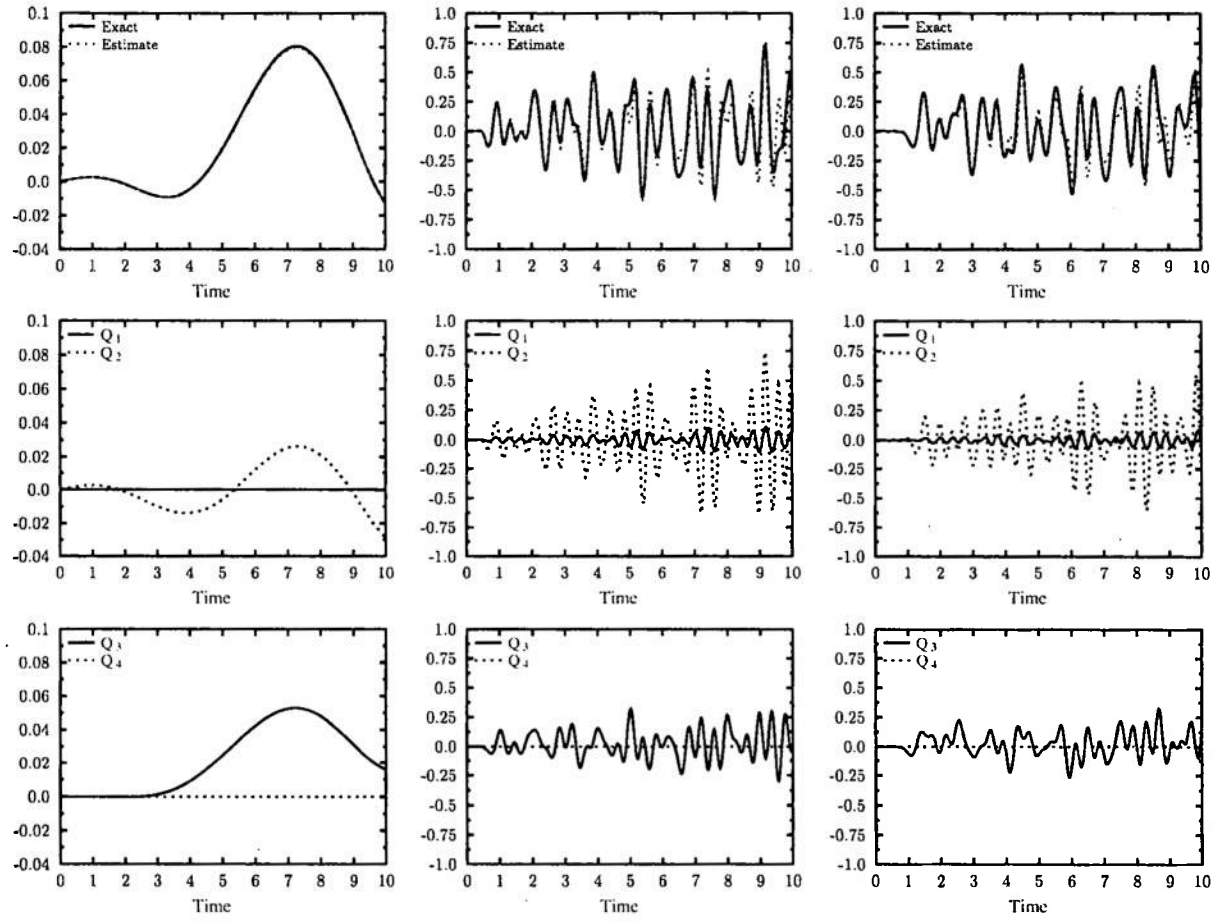


Fig. 13. Time history of error for solving (2) using one iteration. Left: M_1 (slow). Middle: m_3 ("bridge"). Right: m_7 (fast). Time steps: $\Delta t = 0.02$, $\Delta s_1 = \Delta t/32$, $\Delta s_2 = \Delta t$.

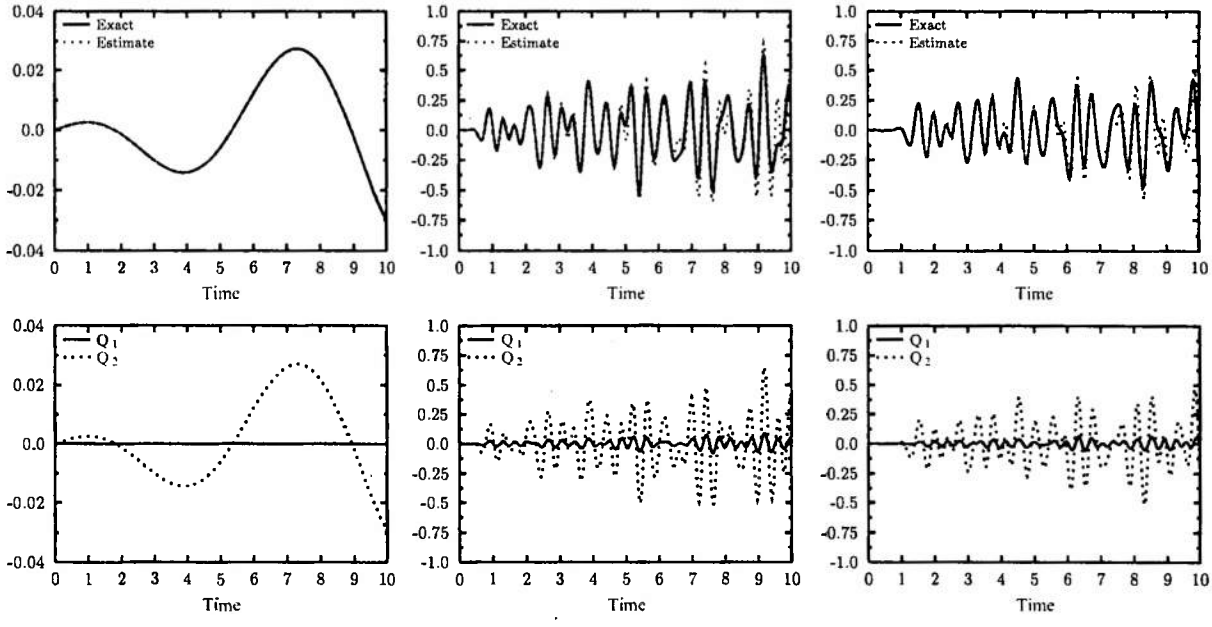


Fig. 14. Time history of error for solving (2) using three iterations. Left: M_1 (slow). Middle: m_3 ("bridge"). Right: m_7 (fast). Q_3 and Q_4 are essentially zero and are not shown. Time steps: $\Delta t = 0.02$, $\Delta s_1 = \Delta t/32$, $\Delta s_2 = \Delta t$.

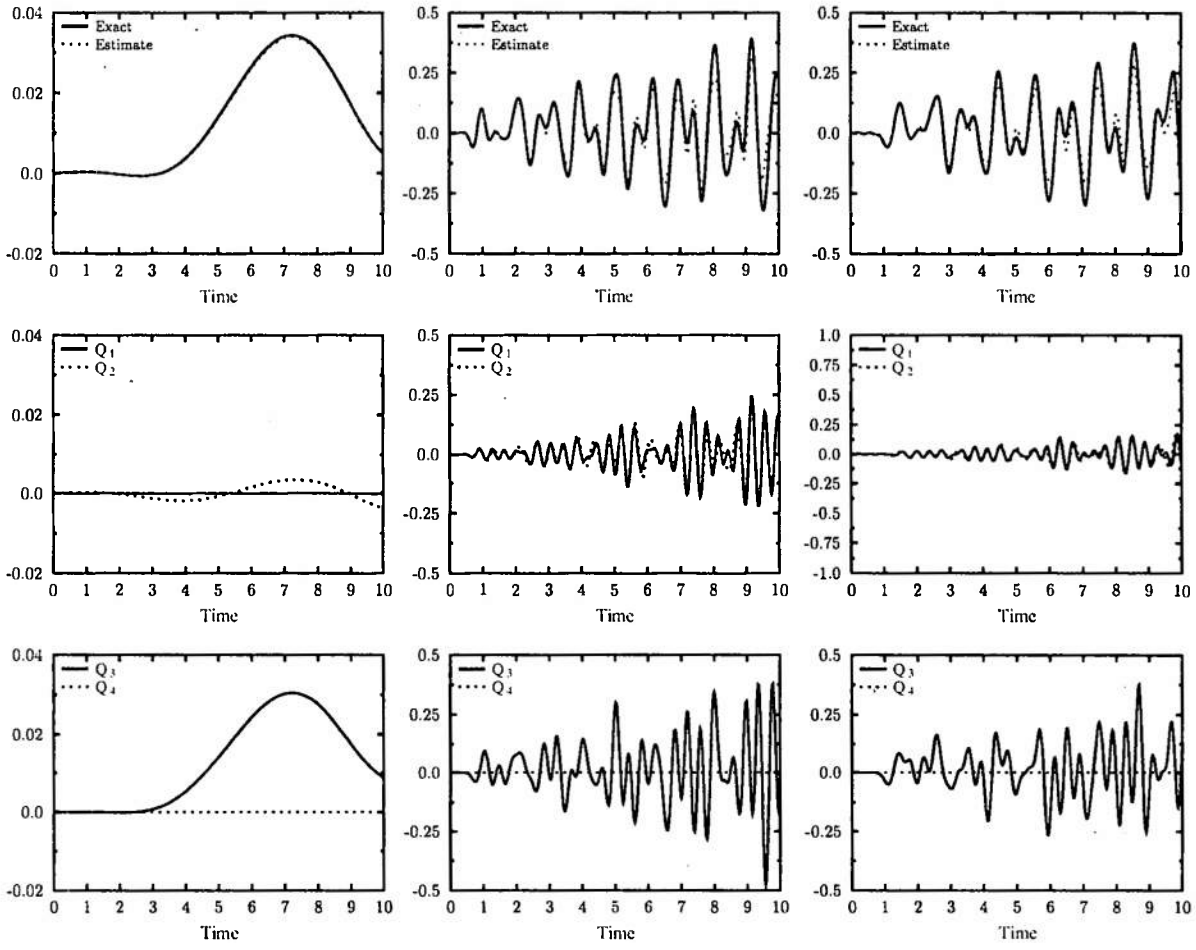


Fig. 15. Time history of error for solving (2) using one iteration. Left: M_1 (slow). Middle: m_3 ("bridge"). Right: m_7 (fast). Time steps: $\Delta t = 0.02$, $\Delta s_1 = \Delta t/32$, $\Delta s_2 = \Delta t/8$.

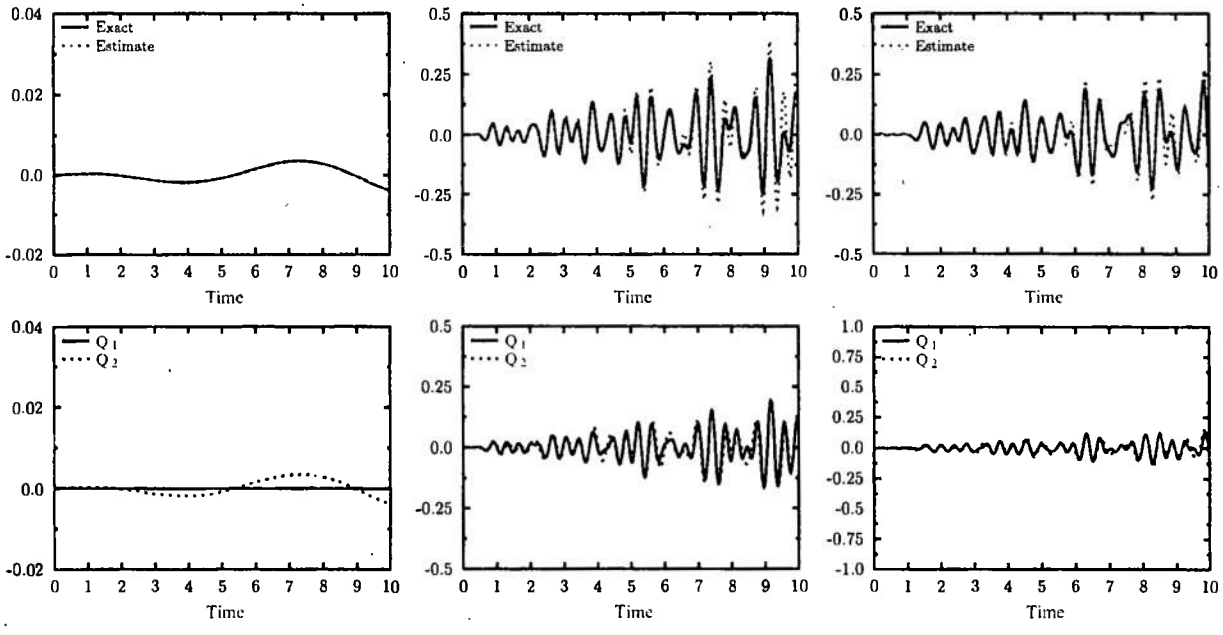


Fig. 16. Time history of error for solving (2) using three iterations. Left: M_1 (slow). Middle: m_3 ("bridge"). Right: m_7 (fast). Q_3 and Q_4 are essentially zero and are not shown. Time steps: $\Delta t = 0.02$, $\Delta s_1 = \Delta t/32$, $\Delta s_2 = \Delta t/8$.

in $C(I_n)$. We pass to the limit in (39), so that this limit satisfies (38). The uniqueness of this limit is established by contradiction using $\Delta t_n \mathcal{L} < 1$. □

Now we can use this lemma to prove Theorem 3.1.

Proof of Theorem 3.1. As in Algorithm 2, $y_1^{(0)} = y_1(t_{n-1})$ and $y_2^{(0)} = y_2(t_{n-1})$. We choose $\Delta t_n < \min\{b/\mathcal{M}, 1/\mathcal{L}\}$, where all the constants are as in the paragraph preceeding Lemma 6.1. The existence of the sequences are established by repeated application of Lemma 6.1. For $m = 1$, we designate $\alpha = y_1(t_{n-1})$ and $\beta = y_2^{(0)}$. Then by Lemma 6.1, the integral equation governing $y_1^{(1)}$ admits a unique solution with $\left(y_1^{(1)}, y_2^{(0)}\right) \in \overline{B}$. Similarly, with $\alpha = y_2(t_{n-1})$ and $\beta = y_1^{(1)}$, Lemma 6.1 guarantees that $y_2^{(1)}$ is unique with $\left(y_1^{(1)}, y_2^{(1)}\right) \in \overline{B}$. We can repeatedly apply this lemma and use induction argument to show that the sequences $\left(y_1^{(m)}, y_2^{(m)}\right) \in \overline{B}$. Our next task is to establish the convergence of the sequences. Note that

$$\left|y_2^{(1)}(t) - y_2^{(0)}(t)\right| = \left|y_2^{(1)}(t) - y_2(t_{n-1})\right| \leqslant \mathcal{M}(t - t_{n-1}) \leqslant \mathcal{M}\Delta t_n.$$

Then by adding and subtracting $F_1\left(y_1^{(1)}, y_2^{(1)}\right)$ and applying the Lipschitz condition for F

$$\begin{aligned}\left|y_1^{(2)}(t) - y_1^{(1)}(t)\right| &\leqslant \int_{t_{n-1}}^{t} \left(\left|F_1\left(y_1^{(2)}, y_2^{(1)}\right) - F_1\left(y_1^{(1)}, y_2^{(1)}\right)\right| \right.\\ &\quad \left. + \left|F_1\left(y_1^{(1)}, y_2^{(1)}\right) - F_1\left(y_1^{(1)}, y_2^{(0)}\right)\right|\right) ds\\ &\leqslant \mathcal{L}\int_{t_{n-1}}^{t} \left|y_1^{(2)}(s) - y_1^{(1)}(s)\right| ds + \frac{1}{2}\,\mathcal{M}\mathcal{L}(t - t_{n-1})^2.\end{aligned}$$

Setting $\tau_n = \mathcal{L}\Delta t_n \exp(\mathcal{L}\Delta t_n)$, we apply a Gronwall's inequality to obtain,

$$\left|y_1^{(2)}(t) - y_1^{(1)}(t)\right| \leqslant \frac{1}{2}\mathcal{M}\mathcal{L}(t - t_{n-1})^2 \exp(\mathcal{L}\Delta t_n) \leqslant \frac{\mathcal{M}}{\mathcal{L}\exp(\mathcal{L}\Delta t_n)}\tau_n^2.$$

Similarly,

$$\left|y_2^{(2)}(t) - y_2^{(1)}(t)\right| \leqslant \frac{\mathcal{M}}{\mathcal{L}\exp(\mathcal{L}\Delta t_n)}\tau_n^2.$$

By induction,

$$\left|y_1^{(m)}(t) - y_1^{(m-1)}(t)\right| \leqslant C_n\tau_n^m \quad \text{and} \quad \left|y_2^{(m)}(t) - y_2^{(m-1)}(t)\right| \leqslant C_n\tau_n^m.$$

where $C_n = \mathcal{M}/(\mathcal{L}\exp(\mathcal{L}\Delta t_n))$. As long as $\tau_n < 1$, we know that for $l > k > N$

$$\begin{aligned}\left|y_1^{(l)}(t) - y_1^{(k)}(t)\right| &\leqslant \sum_{m=k+1}^{l} \left|y_1^{(m)}(t) - y_1^{(m-1)}(t)\right|\\ &\leqslant \sum_{m=N}^{\infty} \left|y_1^{(m)}(t) - y_1^{(m-1)}(t)\right| \leqslant \frac{C_n\tau_n^N}{1 - \tau_n}.\end{aligned}$$

In other words, enforcing $\tau_n < 1$ insures that $y_1^{(m)}(t)$ is a Cauchy sequence of continuous function that converges uniformly to an element in $C(I_n)$. This is also true for $y_2^{(m)}$. We pass to the limit in (9), so that this limit satisfies (1) in I_n. Uniqueness is again established by contradiction. □

6.2. Convergence analysis for the iterative multirate Galerkin finite element method

The errors are denoted by $e_i^{(m)} = y_i^{(m)} - Y_i^{(m)}$ for $i = 1, 2$. It is obvious from the description of the method that the finite element solution $Y^{(m)}$ is a consistent numerical discretization of the differential equations governing $y^{(m)}$. Thus intuitively we expect that the error resulting from this discretization can be bounded by some power of the time steps $\Delta s_{i,n}$. Standard analysis of time discontinuous finite element for solving system of ordinary differential equations have been performed by many authors, see for example [16,17,35,22]. In general, one initiates a local analysis within a time sub-interval under the assumption that the initial condition in this interval is exact. Then some form of recursive formulae is derived which is used to accumulate the contribution from each time sub-interval to yield an error estimate at the final time.

Similar arguments can be employed to bound the errors of $Y_1^{(m)}$ and $Y_2^{(m)}$ separately. However, there are complications that need to be addressed appropriately. Firstly, one or both of the differential equations may be solved by lagging one component an iterate behind the current one, e.g. in Algorithms 1 and 2, we have chosen to lag $Y_2^{(m-1)}$. Secondly, the dependence of function F on both $Y_1^{(m)}$ and $Y_2^{(m)}$ dictates that the accuracy of one component affects the other.

This is reflected in the following two lemmas. Lemma 6.2 compares the numerical solution of component $Y_2^{(m)}$ in time interval I_n with a similar finite element solution with exact initial condition at t_{n-1} (i.e., it is equal to $y_2^{(m)}(t_{n-1})$). As expected, this comparison depends upon both the error in the initial condition at t_{n-1} and on the error in the approximation $Y_1^{(m)}$. Accumulation of local error in each $I_{k,n}$ with $k = 1, \ldots, L_{2,n}$ gives the quantification of accuracy of component $Y_1^{(m)}$. Furthermore, with slight modification of the proof to take account of the fact that the approximate solution Y_2 is known at the previous, rather than the current iteration (at $m-1$ rather than m), a similar estimate is also true for $Y_1^{(m)}$ which we state in Lemma 6.3. In what follows, we set $|u|_{I_{i,n}} = \sup_{t\in I_{i,n}}|u(t)|$, and similarly for I_n.

Lemma 6.2. *Let $Z \in \mathcal{V}^{(q_2)}(I_n)$ satisfy*

$$\int_{I_{k,n}} \left(\dot{Z} - F_2\left(y_1^{(m)}, Z\right), W\right) dt + \left([Z]_{k-1,n}, W_{k-1}^{+}\right) = 0 \tag{40}$$

for all $W \in \mathcal{P}^{(q_2)}(I_{k,n})$, $k = 1, 2, \ldots, L_{2,n}$, and $Z\left(t_{0,n}^{-}\right) = y_2^{(m)}(t_{n-1})$. With $\phi = Z - Y_2^{(m)}$,

$$|\phi|_{I_{k,n}}^2 \leqslant 10\exp(24\mathcal{L}\Delta t_n)\left(\left|e_2^{(m)}(t_{n-1}^{-})\right|^2 + \frac{5\mathcal{L}\Delta t_n}{2}\left|e_1^{(m)}\right|_{I_n}^2\right),$$

for sufficiently small $\Delta s_{2,n}$, where $|e_1^{(m)}|_{I_n} = \max_{1\leqslant k\leqslant L_{2,n}}\left\{|e_1^{(m)}|_{I_{k,n}}\right\}$.

Proof. We know that for $\phi \in \mathcal{P}^{(q_2)}(I_{k,n})$ with $q_2 = 0, 1$,

$$|\phi|_{I_{k,n}}^2 = \max\left\{|\phi_{k-1,n}^{+}|^2, |\phi_{k,n}^{-}|^2\right\} \leqslant |\phi_{k-1,n}^{+}|^2 + |\phi_{k,n}^{-}|^2. \tag{41}$$

Subtracting (6) from (40) gives

$$\int_{I_{k,n}} \left(\dot{\phi} + F_2\left(Y_1^{(m)}, Y_2^{(m)}\right) - F_2\left(y_1^{(m)}, Z\right), W\right) dt + \left([\phi]_{k-1,n}, W_{k-1}^{+}\right) = 0 \tag{42}$$

for every $W \in \mathcal{P}^{(q_2)}(I_{k,n})$. With $W = \phi$ in (42) and using the Lipschitz condition of F, we find that

$$\begin{aligned}\frac{1}{2}|\phi_{k,n}^{-}|^2 + \frac{1}{2}|\phi_{k-1,n}^{+}|^2 &\leqslant \frac{\mathcal{L}}{2}\int_{I_{k,n}} \left(3|\phi|^2 + |e_1^{(m)}|^2\right) dt\\ &\quad + |\phi_{k-1,n}^{-}|\,|\phi_{k-1,n}^{+}|.\end{aligned} \tag{43}$$

This in turn gives

$$|\phi_{k,n}^{-}|^2 \leqslant |\phi_{k-1,n}^{-}|^2 + \mathcal{L}\int_{I_{k,n}} \left(3|\phi|^2 + |e_1^{(m)}|^2\right) dt. \tag{44}$$

Using $W = (t - t_{k-1,n})\dot{\phi}$ in (42) and estimating yield

$$\begin{aligned}\frac{1}{2}\Delta s_{2,n}^2|\dot{\phi}|^2 &\leqslant \mathcal{L}^2\int_{I_{k,n}} \left(|\phi|^2 + |e_1^{(m)}|^2\right)(t - t_{k-1,n})\, dt\\ &\quad + \frac{1}{2}|\dot{\phi}|^2\int_{I_{k,n}} (t - t_{k-1,n})\, dt,\end{aligned}$$

from which we get

$$\frac{1}{4}\Delta s_{2,n}^2|\dot{\phi}|_{I_{k,n}}^2 \leqslant \mathcal{L}^2\Delta s_{2,n}\int_{I_{k,n}}\left(|\phi|^2+\left|e_1^{(m)}\right|^2\right)dt.$$

Because $\int_{I_{k,n}}|\phi|^2dt \leqslant 2\Delta s_{2,n}|\phi_{k,n}^-|^2+\frac{2}{3}\Delta s_{2,n}^3|\dot{\phi}|_{I_{k,n}}^2$, combining this with the last inequality yields

$$\left(1-\frac{2}{3}(2\mathcal{L}\Delta s_{2,n})^2\right)\int_{I_{k,n}}|\phi|^2dt \leqslant 2\Delta s_{2,n}|\phi_{k,n}^-|^2+\frac{2}{3}(2\mathcal{L}\Delta s_{2,n})^2\int_{I_{k,n}}\left|e_1^{(m)}\right|^2dt.$$

Provided that $\left(1-\frac{2}{3}(2\mathcal{L}\Delta s_{2,n})^2\right) > 2/3$ (which is equivalent to having $(2\mathcal{L}\Delta s_{2,n})^2 < 1/2$), we get

$$\begin{aligned}\int_{I_{k,n}}|\phi|^2dt &\leqslant 3\Delta s_{2,n}|\phi_{k,n}^-|^2+(2\mathcal{L}\Delta s_{2,n})^2\int_{I_{k,n}}\left|e_1^{(m)}\right|^2dt\\ &\leqslant 3\Delta s_{2,n}|\phi_{k,n}^-|^2+\frac{1}{2}\int_{I_{k,n}}\left|e_1^{(m)}\right|^2dt. \end{aligned} \tag{45}$$

Substitution of (45) to (44) gives

$$|\phi_{k,n}^-|^2 \leqslant |\phi_{k-1,n}^-|^2+9\mathcal{L}\Delta s_{2,n}|\phi_{k,n}^-|^2+\frac{5}{2}\int_{I_{k,n}}\left|e_1^{(m)}\right|^2dt.$$

Provided that $1-9\mathcal{L}\Delta s_{2,n} > 1/10$ (which is equivalent to having $\mathcal{L}\Delta s_{2,n} < 1/10$), then this last inequality yields

$$|\phi(t_{k,n}^-)|^2 \leqslant \exp(24\mathcal{L}\Delta s_{2,n})\left(|\phi(t_{k-1,n}^-)|^2+\frac{5\mathcal{L}}{2}\int_{I_{k,n}}\left|e_1^{(m)}\right|^2dt\right).$$

We can now undo the recursive relation to get

$$|\phi\left(t_{\bar{k},n}^-\right)|^2 \leqslant \exp(24\mathcal{L}\bar{k}\Delta s_{2,n})\left(|e_2^{(m)}(t_{n-1}^-)|^2+\frac{5\mathcal{L}}{2}\sum_{k=1}^{\bar{k}}\int_{I_{k,n}}\left|e_1^{(m)}\right|^2dt\right) \tag{46}$$

for $\bar{k}=1,2,\ldots,L_{2,n}$. Furthermore, using (41) in (43) and estimating we get

$$|\phi|_{I_{k,n}}^2 \leqslant 4|\phi(t_{k-1,n}^-)|^2+2\mathcal{L}\int_{I_{k,n}}\left(3|\phi|^2+\left|e_1^{(m)}\right|^2\right)dt,$$

which gives

$$(1-6\mathcal{L}\Delta s_{2,n})|\phi|_{I_{k,n}}^2 \leqslant 4|\phi_{k-1,n}^-|^2+2\mathcal{L}\int_{I_{k,n}}\left|e_1^{(m)}\right|^2dt,$$

and thus

$$|\phi|_{I_{k,n}}^2 \leqslant 10|\phi_{k-1,n}^-|^2+5\mathcal{L}\int_{I_{k,n}}\left|e_1^{(m)}\right|^2dt,$$

Using (46) with $\bar{k}=k-1$ in the last inequality, we get

$$\begin{aligned}|\phi|_{I_{k,n}}^2 &\leqslant 10\exp(24\mathcal{L}(k-1)\Delta s_{2,n})\left(\left|e_2^{(m)}(t_{n-1}^-)\right|^2+\frac{5\mathcal{L}}{2}\sum_{j=1}^{k-1}\int_{I_{j,n}}\left|e_1^{(m)}\right|^2dt\right)\\ &\quad+5\mathcal{L}\int_{I_{k,n}}\left|e_1^{(m)}\right|^2dt\\ &\leqslant 10\exp(24\mathcal{L}(k-1)\Delta s_{2,n})\left(\left|e_2^{(m)}(t_{n-1}^-)\right|^2+\frac{5\mathcal{L}}{2}\sum_{j=1}^{k}\int_{I_{j,n}}\left|e_1^{(m)}\right|^2dt\right)\\ &\leqslant 10\exp(24\mathcal{L}\Delta t_n)\left(|e_2^{(m)}(t_{n-1}^-)|^2+\frac{5\mathcal{L}\Delta t_n}{2}\left|e_1^{(m)}\right|_{I_n}^2\right)\quad\square\end{aligned}$$

Lemma 6.3. *Let $X \in \mathcal{V}^{(q_1)}(I_n)$ satisfy*

$$\int_{I_{l,n}}\left(\dot{X}-F_1\left(X,y_2^{(m-1)}\right),V\right)dt+\left([X]_{l-1,n},V_{l-1}^+\right)=0 \tag{47}$$

for all $V \in \mathcal{P}^{(q_1)}(I_{l,n})$, $l=1,2,\ldots,L_{1,n}$, and $X\left(t_{0,n}^-\right)=y_1^{(m)}(t_{n-1})$. With $\xi = X-Y_1^{(m)}$,

$$|\xi|_{I_{l,n}}^2 \leqslant 10\exp(24\mathcal{L}\Delta t_n)\left(|e_1(t_{n-1}^-)|^2+\frac{5\mathcal{L}\Delta t_n}{2}|e_2|_{I_n}^2+\frac{5\mathcal{L}\Delta t_n}{2}\left|r_2^{(m)}\right|_{I_n}^2\right)$$

for sufficiently small $\Delta s_{1,n}$, where $r_2^{(m)}=\left(y_2^{(m)}-y_2^{(m-1)}\right)-\left(Y_2^{(m)}-Y_2^{(m-1)}\right)$.

Proof. This is obtained using the same argument as in Lemma 6.2. □

The two lemmas above are true for any iteration m within a time interval I_n. Not only is it apparent that the accuracy of one component affects the other, but as stated in Lemma 6.3, the error also depends upon the accuracy of the previous iteration. Thus, in addition to the numerical discretization, the iteration residuals would also influence the accuracy of the overall solution. The following lemma states this fact about the error $e^{(m)}=\left[e_1^{(m)}\ e_2^{(m)}\right]^T$.

Lemma 6.4. *For sufficiently small Δt_n, the error of the finite element solutions at iteration level m over time interval I_n satisfies*

$$\begin{aligned}|e^{(m)}(t_n^-)|^2 \leqslant \exp(C\Delta t_n)&\left(|e^{(m)}(t_{n-1}^-)|^2+\Delta t_n^2\left(C_1\Delta s_{1,n}^{2(q_1+1)}+C_2\Delta s_{2,n}^{2(q_2+1)}\right)\right.\\ &\left.+C_3\Delta t_n\left|r_2^{(m)}\right|_{I_n}^2\right).\end{aligned}$$

Proof. Given the finite element solutions $Y_1^{(m)}$ and $Y_2^{(m)}$ over time interval I_n, we use X and Z in Lemmas 6.2 and 6.3, to write

$$e_1^{(m)}=\left(y_1^{(m)}-X\right)+\xi \quad\text{and}\quad e_2^{(m)}=\left(y_2^{(m)}-Z\right)+\phi,$$

where ξ and ϕ are as in those two lemmas. Moreover, it has been established ([17,35,22]) that

$$\left|y_1^{(m)}-X\right|_{I_n} \leqslant C\Delta t_n\Delta s_{1,n}^{q_1+1} \quad\text{and}\quad \left|y_2^{(m)}-X\right|_{I_n} \leqslant C\Delta t_n\Delta s_{2,n}^{q_2+1}.$$

Using Lemmas 6.2 and 6.3, we get

$$\begin{aligned}\left|e_1^{(m)}\right|_{I_n}^2+\left|e_2^{(m)}\right|_{I_n}^2 &\leqslant C\Delta t_n^2\Delta s_{1,n}^{2(q_1+1)}+C\ \Delta t_n^2\Delta s_{2,n}^{2(q_2+1)}\\ &\quad+10\exp(24\mathcal{L}\Delta t_n)\left(\left|e_1^{(m)}(t_{n-1}^-)\right|^2+\frac{5\mathcal{L}\Delta t_n}{2}\left|e_2^{(m)}\right|_{I_n}^2\right.\\ &\quad\left.+\frac{5\mathcal{L}\Delta t_n}{2}\left|r_2^{(m)}\right|_{I_n}^2\right)+10\exp(24\mathcal{L}\Delta t_n)\\ &\quad\times\left(\left|e_2^{(m)}(t_{n-1}^-)\right|^2+\frac{5\mathcal{L}\Delta t_n}{2}\left|e_1^{(m)}\right|_{I_n}^2\right).\end{aligned}$$

Arguing as in Lemma 6.2, for sufficiently small $\Delta\ t_n$, there are constants C, C_1, C_2, and C_3 such that

$$\begin{aligned}\left|e_1^{(m)}\right|_{I_n}^2+\left|e_2^{(m)}\right|_{I_n}^2 &\leqslant \exp(C\Delta t_n)\left(|e^{(m)}(t_{n-1}^-)|^2\right.\\ &\quad\left.+\Delta t_n^2\left(C_1\Delta s_{1,n}^{2(q_1+1)}+C_2\Delta s_{2,n}^{2(q_2+1)}\right)+C_3\Delta t_n|r_2|_{I_n}^2\right).\end{aligned}$$

Because $|e^{(m)}(t_n^-)|^2 \leqslant \left|e_1^{(m)}\right|_{I_n}^2+\left|e_2^{(m)}\right|_{I_n}^2$, this last inequality gives the desired estimate. □

Finally, we have the proof of Theorem 3.2.

Proof. Lemma 6.4 in Section 6 is a recursive relation for the error within time interval I_n. With $m=M_n$, this recursive relation is unwound to obtain the error estimate at the final time $t_N=T$. □

7. Summary

In this paper, we carry out an *a priori* analysis and derive a hybrid *a posteriori – a priori* error estimate for a multirate numerical method for an ordinary differential equation that presents significantly different scales within the components of the model. We formulate an iterative multirate Galerkin finite element method then employ adjoint operators and variational analysis. The *a priori* analysis uses the fact that iterative multirate Galerkin finite element method is a consistent approximation of the analytic fixed point iteration we construct. The hybrid estimate has the form of a computable leading order expression plus uncomputable quantities that are provably higher order in an asymptotic sense. These higher order terms vanish when the convergence in both the solution and adjoint are reached. The computable expression represents the error in terms of contributions from the numerical error arising in the solution of each component, the iteration error, and the error in the adjoint arising from the analytic fixed point iteration. The *a posteriori* analysis takes into account the fact that the original problem and an analytic fixed point iteration are associated with different adjoint problems. We conclude with some examples that demonstrate the accuracy of the computable parts of the hybrid *a posteriori – a priori* estimate.

Acknowledgement

S. Tavener's work is supported in part by the National Science Foundation (UBM-0734267, EF-0914489, DMS-1016268, S-STEM 1060548) and Idaho National Laboratory (00115474).

References

[1] J.F. Andrus, Numerical solution of systems of ordinary differential equations separated into subsystems, SIAM J. Numer. Anal. 16 (1979) 605–611.
[2] J.F. Andrus, Stability of a multi-rate method for numerical integration of ODEs, Comput. Math. Appl. 25 (1993) 3–14.
[3] J.Y. Astic, A. Bihain, M. Jerosolimski, The mixed Adams-BDF variable step size algorithm to simulate transient and long term phenomena in power systems, IEEE Trans. Power Syst. 9 (1994) 929935.
[4] W. Bangerth, R. Rannacher, Adaptive Finite Element Methods for Differential Equations, Birkhauser Verlag, 2003.
[5] A. Bartel, M. Gunther, A multirate W-method for electrical networks in state space formulation, J. Comput. Appl. Math. 147 (2002) 411425.
[6] T.J. Barth, A-posteriori error estimation and mesh adaptivity for finite volume and finite element methods, Lect. Notes Comput. Sci. Engrg., vol. 41, Springer, New York, 2004.
[7] R. Becker, R. Rannacher, An optimal control approach to a posteriori error estimation in finite element methods, Acta Numer. (2001) 1–102.
[8] J.J. Biesiadecki, R. Skeel, Dangers of multiple time step methods, J. Comput. Phys. 109 (1993) 318–328.
[9] V. Carey, D. Estep, S. Tavener, A posteriori analysis and adaptive error control for multiscale operator decomposition solution of elliptic systems I: triangular systems, SINUM 47 (2009) 740–761.
[10] J. Chen, M. Crow, A variable partitioning strategy for the multirate method in power systems, IEEE Trans. Power Syst. 23 (2008) 259–266.
[11] J. Chen, M. Crow, B. Chowdhury, L. Acar, An error analysis of the multirate method for power system transient stability simulation, in: Proceedings IEEE Power Engineering Society Power Systems Conference Expos. vol. 2, 2004, p. 982–986.
[12] E.A. Coddington, N. Levinson, Theory of Ordinary Differential Equations, McGraw-Hill Book Company, Inc., New York, 1955.
[13] E. Constantinescu, A. Sandu, Multirate timestepping methods for hyperbolic conservation laws, 2006, Report TR-06-15.
[14] M. Crow, J.G. Chen, The multirate method for simulation of power system dynamics, IEEE Trans. Power Syst. 9 (1994) 16841690.
[15] C. Dawson, R. Kirby, High resolution schemes for conservation laws with locally varying time steps, SIAM J. Sci. Comput. 22 (2000) 2256–2281.
[16] M. Delfour, W. Hager, F. Trochu, Discontinuous Galerkin methods for ordinary differential equations, Math. Comput. 36 (1981) 455–473.
[17] M.C. Delfour, F. Dubeau, Discontinuous polynomial approximations in the theory of one-step, hybrid and multistep methods for nonlinear ordinary differential equations, Math. Comput. 47 (1986) 169–189. S1–S8, With a supplement.
[18] C. Engstler, C. Lubich, A multirate extension of the eight-order Dormand–Prince method, Appl. Numer. Math. 25 (1997) 185–192.
[19] C. Engstler, C. Lubich, Multirate extrapolation methods for differential equations with different time scales, Computing 58 (1997) 173–185.
[20] K. Eriksson, D. Estep, P. Hansbo, C. Johnson, Introduction to Adaptive Methods for Differential Equations, Acta numerica, Cambridge University Press, Cambridge, 1995.
[21] K. Eriksson, D. Estep, P. Hansbo, C. Johnson, Computational Differential Equations, Cambridge University Press, Cambridge, 1996.
[22] D. Estep, A posteriori error bounds and global error control for approximation of ordinary differential equations, SIAM J. Numer. Anal. 32 (1995) 1–48.
[23] D. Estep, Error estimation for multiscale operator decomposition for multiphysics problems, in: J. Fish (Ed.), Bridging the Scales in Science and Engineering, Oxford University Press, 2008.
[24] D. Estep, D. French, Global error control for the continuous Galerkin finite element method for ordinary differential equations, RAIRO Modél. Math. Anal. Numér. 28 (1994) 815–852.
[25] D. Estep, V. Ginting, D. Ropp, J. Shadid, S. Tavener, An a posteriori-a priori analysis of multiscale operator splitting, SIAM J. Numer. Anal. 46 (2008) 1116–1146.
[26] D. Estep, M.G. Larson, R.D. Williams, Estimating the error of numerical solutions of systems of reaction-diffusion equations, Mem. Amer. Math. Soc. 146 (2000) viii+109.
[27] D. Estep, S. Tavener, T. Wildey, A posteriori analysis and improved accuracy for an operator decomposition solution of a conjugate heat transfer problem, SINUM 46 (2008) 2068–2089.
[28] D. Estep, S. Tavener, T. Wildey, A posteriori error estimation and adaptive mesh refinement for a multi-discretization operator decomposition approach to fluid–solid heat transfer, J. Comput. Phys. 229 (2010) 4143–4158.
[29] C. Gear, Multirate methods for ordinary differential equations, 1974, Technical Report #880, University of Illinois at Urbana-Champaign.
[30] C. Gear, D. Wells, Multirate linear multistep methods, BIT 24 (1984).
[31] M. Giles, E. Süli, Adjoint methods for PDEs: a posteriori error analysis and postprocessing by duality, Acta Numer. (2002) 145–236.
[32] M. Gunther, A. Kvaern, P. Rentrop, Multirate partitioned Runge–Kutta methods, BIT 41 (2001) 504514.
[33] M. Gunther, P. Rentrop, Partitioning and multirate strategies in latent electric circuits, Int. Ser. Numer. Math. 117 (1994) 3360.
[34] W. Hundsdorfer, V. Savcenco, Analysis of a multirate theta-method for stiff ODEs, Appl. Numer. Math. 59 (2009) 693–706.
[35] C. Johnson, Error estimates and adaptive time-step control for a class of one-step methods for stiff ordinaty differential equations, SIAM J. Numer. Anal. 25 (1988) 908–926.
[36] R. Kirby, On the convergence of of high resolution methods with multiple time scales for hyperbolic conservation laws, Math. Comput. 72 (2003) 1239–1250.
[37] A. Kurita, H. Okubo, K. Oki, S. Agematsua, D. Klapper, N. Miller, J. Price, J. Sanchez-Gasca, K. Wirgau, T. Younkins, Multiple time-scale power system dynamic simulation, IEEE Trans. Power Syst. 8 (1993) 216223.
[38] A. Kvaerno, Stability of multirate Runge–Kutta schemes, Int. J. Differ. Equ. Appl. 1A (2000) 97–105.
[39] C. Lanczos, Linear Differential Operators, Dover Publications, 1997.
[40] A. Logg, Multi-adaptive Galerkin methods for ODEs I, SIAM J. Scient. Comput. 24 (2003) 1879–1902.
[41] A. Logg, Multi-adaptive Galerkin methods for ODEs. II. Implementation and applications, SIAM J. Sci. Comput. 25 (2003/04) 1119–1141. electronic.
[42] A. Logg, Multiadaptive Galerkin methods for ODEs. III. A priori error estimates, SIAM J. Numer. Anal. 43 (2006) 2624–2646. electronic.
[43] G.I. Marchuk, V.I. Agoshkov, V.P. Shutyaev, Adjoint Equations and Perturbation Algorithms in Nonlinear Problems, CRC Press, Boca Raton, FL, 1996.
[44] N. Maurits, H. van der Ven, A. Veldman, Explicit multi-time stepping methods for convection dominated flow problems, Comput. Meth. Appl. Mech. Engrg. 157 (1998) 133–150.
[45] S. Osher, R. Sanders, Numerical approximations to nonlinear conservation laws with locally varying time and space grids, Math. Comput. 41 (1983) 321–336.
[46] S.D. Pekarek, O. Wasynczuk, E. Walters, J. Jatskevich, C. Lucas, N. Wu, P. Lamm, An efficient multirate simulation technique for power-electronic-based systems, IEEE Trans. Power Syst. 19 (2004) 399409.
[47] L. Perko, Differential Equations and Dynamical Systems, Springer-Verlag, New York, 2001.
[48] J.R. Rice, Split Runge–Kutta methods for simultaneous equations, J. Res. Natl. Bur. Standards 64B (1960) 151–170.
[49] G. Rodriguez-Gomez, P. Gonzalez-Casanova, J. Martinez-Carballido, Computing general companion matrices and stability regions of multirate methods, Int. J. Numer. Methods Engrg. 61 (2004) 255–273.
[50] J.J. Sanchez-Gasca, R. DAquila, J. Paserba, W. Price, D. Klapper, I.P. Hu, Extended-term dynamic simulation using variable time step integration, IEEE Comput. Appl. Power 6 (1993) 2328.
[51] J. Sand, S. Skelboe, Stability of backward Euler multirate methods and convergence of waveform relaxation, BIT 32 (1992) 350–366.
[52] V. Savcenco, Comparison of the asymptotic stability properties for two multirate strategies, J. Comput. Appl. Math. 220 (2008) 508–524.
[53] V. Savcenco, Construction of a multirate RODAS method for stiff ODEs, J. Comput. Appl. Math. 225 (2009) 323–337.
[54] V. Savcenco, W. Hundsdorfer, J. Verwer, A multirate time stepping strategy for stiff ordinary differential equations, BIT 47 (2007) 137155.
[55] S. Skelboe, Stability properties of backward differentiation multirate formulas, Appl. Numer. Math. 5 (1989) 151–160.

[S6] S. Skelboe, P.U. Anderson, Stability properties of backward Euler multirate formulas, SIAM J. Sci. Stat. Comput. 10 (1989) 1000–1009.
[S7] M. Striebel, M. Gunther, A charge oriented mixed multirate method for a special class of index – 1 network equations in chip design, Appl. Numer. Math. 53 (2005) 489–507.
[S8] H. Tang, G. Warnecke, High resolution schemes for conservation laws and convection–diffusion equations with varying time and space grids, J. Comput. Math. 24 (2006) 121–140.
[S9] V. Thomée, Galerkin Finite Element Methods for Parabolic Problems, Springer-Verlag, Berlin, 1997.
[60] A. Verhoeven, A.E. Guennouni, E. ter Maten, R. Mattheij, A general compound multirate method for circuit simulation problems, in: Scientific Computing in Electrical Engineering, Springer, 2006, p. 1431S0.
[61] A. Verhoeven, B. Tasic, T.G.J. Beelen, E.J.W. ter Maten, R.M.M. Mattheij, Automatic partitioning for multirate methods, Scientific Computing in Electrical Engineering, vol. 11, Springer, Berlin, 2007.

INTERNATIONAL JOURNAL FOR NUMERICAL METHODS IN ENGINEERING
Int. J. Numer. Meth. Engng 2010; **00**:1–24
Published online in Wiley InterScience (www.interscience.wiley.com). DOI: 10.1002/nme

A posteriori analysis and adaptive error control for operator decomposition solution of coupled semilinear elliptic systems

V. Carey[1], D. Estep[2*] and S. Tavener[1]

[1]*Department of Mathematics, Colorado State University, Fort Collins, CO 80523*
[2]*Department of Statistics, Colorado State University, Fort Collins, CO 80523*

SUMMARY

In this paper, we develop an *a posteriori* error analysis for operator decomposition iteration methods applied to systems of coupled semilinear elliptic problems. The goal is to compute accurate error estimates that account for the combined effects arising from numerical approximation (discretization) and operator decomposition iteration. In an earlier paper [1], we considered "triangular" systems that can be solved without iteration. In contrast, operator decomposition iterative methods for fully coupled systems involve an iterative solution technique. We construct an error estimate for the numerical approximation error that specifically addresses the propagation of error between iterates and provide a computable estimate for the iteration error arising due to the decomposition of the operator. Finally, we extend the adaptive discretization strategy in [1] to systematically reduce the discretization error. Copyright © 2010 John Wiley & Sons, Ltd.

Received ...

KEY WORDS: a posteriori error estimates, adjoint problem, dual problem, error estimates, finite element method, generalized Green's function, operator splitting, operator decomposition, coupled problems

1. INTRODUCTION

We develop an *a posteriori* error analysis framework for operator decomposition iteration methods applied to systems of coupled semilinear elliptic problems of the form,

$$\begin{aligned}
\mathcal{L}_1(x,u_1,Du_1,D^2u_1) &= f_1(x,u_1,u_2,Du_2,u_3,Du_3,\cdots,u_n,Du_n), \quad x\in\Omega,\\
\mathcal{L}_2(x,u_2,Du_2,D^2u_2) &= f_2(x,u_1,Du_1,u_2,u_3,Du_3,\cdots,u_n,Du_n), \quad x\in\Omega,\\
&\vdots\\
\mathcal{L}_n(x,u_n,Du_n,D^2u_n) &= f_n(x,u_1,Du_1,u_2,Du_2,\cdots,u_{n-1},Du_{n-1},u_n), \quad x\in\Omega,
\end{aligned} \tag{1.1}$$

where Dv and D^2v are the first and second order derivative operators, $\{\mathcal{L}_i, i=1,\cdots,n\}$, is a collection of linear uniformly coercive, elliptic differential operators, $\{f_i, i=1,\cdots,n\}$ is a

*Correspondence to: Department of Statistics, Colorado State University, Fort Collins, CO 80523. E-mail: estep@stat.colostate.edu

Contract/grant sponsor: Defense Threat Reduction Agency (HDTRA1-09-1-0036), Department of Energy (DE-FG02-04ER25620, DE-FG02-05ER25699, DE-FC02-07ER54909, DE-SC0001724), Lawrence Livermore National Laboratory (B573139, B584647), National Aeronautics and Space Administration (NNG04GH63G), National Science Foundation (DMS-1065046, DMS-0107832, DMS-0715135, DGE-0221595003, MSPA-CSE-0434354, ECCS-0700559), Idaho National Laboratory (00069249, 00115474), National Institutes of Health (R01GM096192), Sandia Corporation (PO299784)

collection of differentiable functions, Ω is a convex polygonal domain with boundary $\partial\Omega$, and (1.1) is supplied with suitable boundary conditions on $\partial\Omega$. We note that the coupling in the system occurs through the right-hand-sides only. We assume that (1.1) satisfies suitable conditions to guarantee a solution in $\tilde{W}_2^1(\Omega)$ in weak form, e.g. generic conditions involve a uniform bound on the derivatives of f. An extension of our analysis to fully nonlinear elliptic systems is straightforward but tedious in detail, e.g. involving a messy linearization of the diffusion operator.

Interest in coupled physics problems and their solution arises in many fields. The Oregonator model for the Belousov-Zhabotinsky reaction system,

$$\begin{aligned} \epsilon\frac{\partial u_1}{\partial t} - \epsilon D_1\nabla^2 u_1 &= u_1 - u_1^2 - fu_2\frac{u_1 - q}{u_1 + q}, \\ \frac{\partial u_2}{\partial t} - D_2\nabla^2 u_2 &= u_1 - u_2, \end{aligned}$$

is an example of an important time-dependent coupled semilinear system. In order to consider waves traveling with permanent form with velocity c in the x-direction, we make the ansatz, $u_i(t,x,y) = u_i(\eta,y), i = 1,2$, where $\eta = x - ct$. Upon substitution we obtain the stationary coupled semilinear elliptic system,

$$\begin{aligned} -\epsilon D_1\nabla_*^2 u_1 &= \epsilon c\frac{\partial u_1}{\partial \eta} + u_1 - u_1^2 - fu_2\frac{u_1 - q}{u_1 + q}, \\ -D_2\nabla_*^2 u_2 &= c\frac{\partial u_2}{\partial \eta} + u_1 - u_2, \end{aligned}$$

where $\nabla_*^2(\cdot) = \partial^2(\cdot)/\partial\eta^2 + \partial^2(\cdot)/\partial y^2$.

In many practical situations, coupled systems of partial differential equations are decomposed into individual physics components, each of which is solved with a code specialized to the particular type of physics, while the solution is obtained by various forms of iteration and/or operator decomposition. Such approaches introduce new forms of instability and new sources of error that must be included in an *a posteriori* error analysis, see [2] for an overview.

In this paper, we assume that the system (1.1) is solved by an operator decomposition approach that involves iteratively solving for u_i, $i = 1,\ldots n$, the ordered sequence of problems

$$\mathcal{L}_i(x,u_i,Du_i,D^2u_i) = f_i(x,\hat{u}_1,D\hat{u}_1,\hat{u}_2,D\hat{u}_2,\cdots,\hat{u}_{i-1},D\hat{u}_{i-1},u_i,\hat{u}_{i+1},D\hat{u}_{i+1},\cdots,\hat{u}_n,D\hat{u}_n),$$

which are obtained by substituting solutions $\hat{u}_j$ for $j \neq i$ computed in a previous step in the equations in (1.1) and then solving for u_i. Theoretically, the sequence is iterated until convergence (if it does converge), while in practice, a finite number of iterations is used.

In [1], we present an *a posteriori* error analysis for operator decomposition methods applied to systems of elliptic equations that had an "upper triangular" form, so that one iteration through the system produces the solution. That analysis accounts for:

- errors arising from the discretization of each component elliptic problem,
- the transfer of error between the component elliptic problems,
- errors resulting from using different discretizations for the component elliptic problems.

In a fully coupled system (1.1), we need to iterate through the components until, hopefully, the iteration converges. In addition to the errors affecting the solution of a triangular system, the *a posteriori* error analysis now must also account for:

- the effects of finite iteration,
- the transfer of errors between iterations.

These two effects are the focus of the analysis in this paper. The results in this paper can be combined with the full analysis of [1] to treat all five of these effects in one estimate.

We let $U^{\{k\}}$ be a numerical approximation obtained by iterating numerical discretizations of the differential equations k times. To carry out the *a posteriori* error analysis, we decompose the error

DOI: 10.1002/nme

as

$$u - U^{\{k\}} = \underbrace{u - u^{\{k\}}}_{\text{analytic iteration error}} + \underbrace{u^{\{k\}} - U^{\{k\}}}_{\text{numerical error}} = \mathcal{E}^{\{k\}} + e^{\{k\}}, \tag{1.2}$$

where $u^{\{k\}}$ is the analytic solution obtained at iteration k by solving the sequence of iterated differential equations exactly. We estimate these two components separately. This decomposition is motivated by the observation that the iterative discrete approximation is a consistent numerical solution of the analytic iterative problem. One consequence is that this simplifies the definition of an appropriate adjoint for the error analysis. Namely, we use the *adjoint associated with the solution operator of the sequence of iterated component problems*. This is a complex operator that can itself be defined through an iterated sequence of adjoint problems to the individual components. In practical terms, we can compute the resulting *a posteriori* error estimate *without* forming and solving the adjoint for the fully coupled system. Solving the full adjoint of the coupled system is computationally impractical in situations in which operator decomposition iteration is used to solve the forward problem.

The main focus of this paper is *a posteriori* error estimation of the numerical error $e^{\{k\}}$. At the kth step of an iterative solution process and for a given quantity of interest, the analysis accounts for:

- the numerical errors made at the current iteration,
- the numerical errors made at all previous iterations,
- the error due to the iterative approximation.

For clarity of exposition, we do not include treatment of the errors arising from the use of different discretizations for different components. Such effects are already treated in the earlier paper [1] and those results can be combined with the results in this paper in a straightforward way.

To obtain a full *a posteriori* error estimate of the error $u - U^{\{k\}}$, we have to also estimate the analytic iteration error $\mathcal{E}^{\{k\}}$. The difficulty is that this involves the true solution and an "iterative" true solution, both of which are unknown. However, for a fixed space mesh, we can adapt the classical asymptotic estimator for the error in an iterative approximation to this situation. Lacking such an estimate, the *a posteriori* error estimate of $e^{\{k\}}$ provides an estimate of the full error provided the numerical error dominates the iteration error, i.e. in the limit of increasing iterations.

When we have estimates for $\mathcal{E}^{\{k\}}$ and $e^{\{k\}}$, we can then derive a generalized adaptive algorithm that adjusts both the number of iterations and the level of discretization in each component to achieve a desired accuracy with relative computational efficiency.

This work can be differentiated from a number of previous analyses of nonlinear problems, e.g. see references in [3–9], in concentrating on new issues arising in fully coupled systems and dealing explicitly with the effects of operator decomposition and finite iteration. Ignoring the coupling involved in treating a system, e.g. simply estimating the error in each component in isolation, can lead to catastrophic failure of the error estimates. See Example 3.2 of [1] and Example 4.1 below. The alternative approach, which uses the adjoint of the fully coupled semilinear elliptic system, provides a valid error estimate (up to linearization error), but fails to differentiate the sources of error.

The outline of the paper is as follows. To explain the ideas behind the definition of an appropriate adjoint operator for the iterated system and the *a posteriori* error analysis, we begin by deriving an *a posteriori* error estimate for the iterated solution of a finite dimensional algebraic systems in Section 2. This derivation contains the main ideas without the complications of differential equation discretization errors and moreover makes it easy to construct several illuminating examples. We then turn to the analysis of coupled semilinear elliptic problems in Section 3. We present results for three particular classes of operator decomposition iteration techniques, namely block Jacobi, block Gauss-Seidel and relaxed block Jacobi. In Section 4, we give numerical examples of different aspects of the error estimation framework, concluding with an adaptive algorithm that adaptively refines both the computational mesh and the operator decomposition iteration to converge to an accurate solution.

DOI: 10.1002/nme

2. PRELIMINARY EXAMPLE: ITERATIVE SOLUTION OF ALGEBRAIC SYSTEMS

In order to explain the construction of an appropriate adjoint operator for an iterated solution approach and the idea that the numerical error of the current iterate is affected by errors introduced at all previous iterations, we first present the analysis in the context of finite dimensional algebraic systems. We begin with a linear problem and then treat a nonlinear problem.

2.1. Estimating the numerical error for linear algebraic systems

We consider the solution of

$$\mathbf{A}w = b.$$

We construct an iterative solution method using a matrix decomposition of $\mathbf{A}$ of the form

$$\mathbf{A} = \mathbf{D} + \mathbf{C},$$

where we assume that $\mathbf{D}$ is invertible. The solution procedure uses the observation that

$$(\mathbf{D} + \mathbf{C})w = b \Rightarrow \mathbf{D}w = b - \mathbf{C}w$$

and solves

$$\left.\begin{aligned} w^{\{0\}} &= 0, \\ \mathbf{D}w^{\{i\}} &= b - \mathbf{C}w^{\{i-1\}},\ i = 1, 2, \ldots, k \end{aligned}\right\}. \tag{2.1}$$

We assume that the iterative scheme converges, which depends on the spectral radius of $\mathbf{D}^{-1}\mathbf{C}$.

At each stage of the iterative process, we compute a numerical solution $W^{\{i\}} \approx w^{\{i\}}$. Our goal is to estimate the error in a quantity of interest that is representable as a linear functional of the solution, i.e., a quantity of interest of the form (w, ψ), at any iterate i. We write this error as

$$\left(w - W^{\{i\}}, \psi\right) = \underbrace{\left(w - w^{\{i\}}, \psi\right)}_{\text{analytic iteration error}} + \underbrace{\left(w^{\{i\}} - W^{\{i\}}, \psi\right)}_{\text{numerical error}} = \mathcal{E}^{\{i\}} + e^{\{i\}}.$$

Here, a superscript in braces $^{\{i\}}$ indicates variables corresponding to forward iteration i. Let k be the total number of forward iterations performed. Later, we use a superscript in square brackets $^{[j]}$ to denote the adjoint problem corresponding to forward iteration $i = k - j + 1$.

Finally, we introduce the notation

$$\mathcal{R}_i(U^{\{j\}}, \phi^{[k]}; U^{\{j-1\}})$$

to denote the residual of equation i, at iteration j, weighted by the kth adjoint problem, evaluated using the solution generated at iteration $j - 1$.

2.1.1. Estimation of the numerical error $e^{\{i\}}$. Because the previous iterate enters as data, the numerical error $e^{\{i\}}$ depends not only upon the numerical error made at the i^{th} iteration, but also on the numerical errors made at previous iterations. Hence, we need to estimate the effects of these "inherited errors".

Given $W^{\{0\}} = 0$, we compute the sequence of approximate solutions $W^{\{i\}}$ of (2.1), $i = 1, \ldots, k$, as

$$\mathbf{D}W^{\{i\}} = b - \mathbf{C}W^{\{i-1\}}. \tag{2.2}$$

Theorem 2.1
The error in the quantity of interest can be estimated as,

$$(e^{\{k\}}, \psi) = \sum_{j=1}^{k} \mathcal{R}\left(W^{\{k-j+1\}}, \phi^{[j]}; W^{\{k-j\}}\right), \tag{2.3}$$

DOI: 10.1002/nme

where the adjoint problems are defined,

$$\mathbf{D}^{\top}\phi^{[1]} = \psi, \tag{2.4}$$
$$\mathbf{D}^{\top}\phi^{[j]} = -\mathbf{C}^{\top}\phi^{[j-1]}, \quad j = 2, \ldots, k, \tag{2.5}$$

and

$$\mathcal{R}\big(W^{\{k\}}, \phi^{[j]}; W^{\{m\}}\big) = \big(b - \mathbf{C}W^{\{m\}} - \mathbf{D}W^{\{k\}}, \phi^{[j]}\big). \tag{2.6}$$

Note that the adjoint problems (2.4)-(2.5) involve the simpler matrix associated with the iterative scheme, not the full adjoint of the original matrix A. To emphasize the role of the error in the most recent (kth) iteration, we may write

$$(e^{\{k\}}, \psi) = \big(W^{\{k\}}, \phi^{[1]}; W^{\{k-1\}}\big) + \sum_{j=2}^{k} \mathcal{R}\big(W^{\{k-j+1\}}, \phi^{[j]}; W^{\{k-j\}}\big). \tag{2.7}$$

Also note that the sequence of coupled adjoint problems (2.5) yield the natural adjoint to the solution operator associated with the iterative method. We emphasize this point by numbering the adjoint problems in the reverse order to the forward iteration number.

Proof

The estimates of the numerical error after k iterations is

$$\begin{aligned}
\big(e^{\{k\}}, \psi\big) &= \big(w^{\{k\}} - W^{\{k\}}, \mathbf{D}^{\top}\phi^{[1]}\big) = \big(\mathbf{D}(w^{\{k\}} - W^{\{k\}}), \phi^{[1]}\big) \\
&= \big(b - \mathbf{C}w^{\{k-1\}} - \mathbf{D}W^{\{k\}}, \phi^{[1]}\big) \\
&= \big(b - \mathbf{C}W^{\{k-1\}} - \mathbf{D}W^{\{k\}}, \phi^{[1]}\big) - \big(\mathbf{C}(w^{\{k-1\}} - W^{\{k-1\}}), \phi^{[1]}\big) \\
&= \mathcal{R}\big(W^{\{k\}}, \phi^{[1]}; W^{\{k-1\}}\big) - \big(\mathbf{C}(w^{\{k-1\}} - W^{\{k-1\}}), \phi^{[1]}\big).
\end{aligned} \tag{2.8}$$

The first term on the right of (2.8) is the usual error estimate depending on the computable residual of $W^{\{k\}}$ and the corresponding adjoint solution. The second term on the right of (2.8) estimates the contribution to the error of $W^{\{k\}}$ resulting from the error inherited from the approximation $W^{\{k-1\}}$ to $w^{\{k-1\}}$. This *inherited error* term can be expressed as the error in a *new* quantity of interest at the previous iteration by noting that

$$-\big(\mathbf{C}(w^{\{k-1\}} - W^{\{k-1\}}), \phi^{[1]}\big) = \big(w^{\{k-1\}} - W^{\{k-1\}}, -\mathbf{C}^{\top}\phi^{[1]}\big) = \big(e^{\{k-1\}}, -\mathbf{C}^{\top}\phi^{[1]}\big).$$

To estimate the error in this new quantity of interest, we solve the new adjoint problem

$$\mathbf{D}^{\top}\phi^{[2]} = -\mathbf{C}^{\top}\phi^{[1]},$$

to obtain

$$\begin{aligned}
\big(e^{\{k-1\}}, -\mathbf{C}^{\top}\phi^{[1]}\big) &= \big(w^{\{k-1\}} - W^{\{k-1\}}, -\mathbf{C}^{\top}\phi^{[1]}\big) = \big(w^{\{k-1\}} - W^{\{k-1\}}, \mathbf{D}^{\top}\phi^{[2]}\big) \\
&= \big(b - \mathbf{C}w^{\{k-2\}} - \mathbf{D}W^{\{k-1\}}, \phi^{[2]}\big) \\
&= \big(b - \mathbf{C}W^{\{k-2\}} - \mathbf{D}W^{\{k-1\}}, \phi^{[2]}\big) - \big(\mathbf{C}(w^{\{k-2\}} - W^{\{k-2\}}), \phi^{[2]}\big) \\
&= \mathcal{R}\big(W^{\{k-1\}}, \phi^{[2]}; W^{\{k-2\}}\big) - \big(\mathbf{C}(w^{\{k-2\}} - W^{\{k-2\}}), \phi^{[2]}\big).
\end{aligned}$$

Continuing in this fashion, we see that the desired estimate of the error in the quantity of interest $(e^{\{k\}}, \psi)$ after k iterations requires the solution of (2.4) and the recursive solution of an *ordered* sequence of $(k-1)$ adjoint problems

$$\mathbf{D}^{\top}\phi^{[j]} = -\mathbf{C}^{\top}\phi^{[j-1]}, \quad j = 2, \ldots, k,$$

and

$$\big(e^{\{k\}}, \psi\big) = \sum_{j=1}^{k} \mathcal{R}\big(W^{\{k-j+1\}}, \phi^{[j]}; W^{\{k-j\}}\big).$$

□

DOI: 10.1002/nme

2.1.2. The decay of influence of contributions from early iterations. To estimate the numerical error in the quantity of interest after k iterations, we nominally need to solve k adjoint problems. (One of the form (2.4) and $k-1$ of the form (2.5)). Each of these is solvable, but the number can be significant. The sequence of adjoint problems has the form

$$\phi^{[j+1]} = -(\mathbf{D}^{-\mathsf{T}}\mathbf{C}^{\mathsf{T}})^j \phi^{[1]}, \quad j = 1, \ldots, k-1.$$

By noting that

$$\mathbf{D}^{-1}(\mathbf{C}\mathbf{D}^{-1})\mathbf{D} = \mathbf{D}^{-1}\mathbf{C},$$

we see that $\mathbf{D}^{-1}\mathbf{C}$, $\mathbf{C}\mathbf{D}^{-1}$ and $\mathbf{D}^{-\mathsf{T}}\mathbf{C}^{\mathsf{T}}$ all have the same spectral radius. For the forward iteration to converge the matrix product $\mathbf{D}^{-1}\mathbf{C}$ must have a spectral radius smaller than one. This means that we expect that

$$\phi^{[m-l+1]} = -(\mathbf{D}^{-\mathsf{T}}\mathbf{C}^{\mathsf{T}})^{m-l}\phi^{[1]} \to 0$$

for fixed l as $m \to \infty$. This suggests that we might obtain a reasonable approximation

$$\left(e^{\{k\}}, \psi\right) \approx \sum_{j=1}^{l} \mathcal{R}\left(W^{\{k-j+1\}}, \phi^{[j]}; W^{\{k-j\}}\right) \tag{2.9}$$

for some small $l \geq 1$. The more rapid the convergence of the forward iteration (the smaller the spectral radius of $\mathbf{D}^{-1}\mathbf{C}$), the smaller the value of l required.

2.1.3. Estimating the iterative error $\mathcal{E}^{\{k\}}$. We begin with a classic estimate for the iteration error based on extrapolation. We define $\mathcal{J}(\cdot) = (\cdot, \psi)$ and denote $\mathcal{E}^{\{k\}} = w - w^{\{k\}}$. Assuming a linearly convergent sequence $\mathcal{J}\left(\mathcal{E}^{\{i\}}\right)$, the classic asymptotic argument yields,

$$\mathcal{J}\left(\mathcal{E}^{\{k\}}\right) \approx -\frac{\left(\mathcal{J}\left(w^{\{k\}}\right) - \mathcal{J}\left(w^{\{k-1\}}\right)\right)^2}{\mathcal{J}\left(w^{\{k\}}\right) - \mathcal{J}\left(w^{\{k-1\}}\right) - \left(\mathcal{J}\left(w^{\{k-1\}}\right) - \mathcal{J}\left(w^{\{k-2\}}\right)\right)}.$$

This estimate is not directly computable, however, the righthand side can be further estimated as,

$$\mathcal{J}\left(\mathcal{E}^{\{k\}}\right) \approx -\frac{\left(\mathcal{J}\left(e^{\{k\}}\right) - \mathcal{J}\left(e^{\{k-1\}}\right) + \mathcal{J}\left(W^{\{k\}}\right) - \mathcal{J}\left(W^{\{k-1\}}\right)\right)^2}{\mathcal{J}\left(e^{\{k\}}\right) - 2\mathcal{J}\left(e^{\{k-1\}}\right) + \mathcal{J}\left(e^{\{k-2\}}\right) + \mathcal{J}\left(W^{\{k\}}\right) - 2\mathcal{J}\left(W^{\{k-1\}}\right) + \mathcal{J}\left(W^{\{k-2\}}\right)}. \tag{2.10}$$

The expression (2.10) can be estimated using the *a posteriori* error estimate, but this is expensive. We may derive another estimate by assuming the numerical error is higher-order than the iteration error, to obtain

$$\mathcal{J}\left(\mathcal{E}^{\{k\}}\right) \approx \frac{-\left(\mathcal{J}\left(W^{\{k\}}\right) - \mathcal{J}\left(W^{\{k-1\}}\right)\right)^2}{\mathcal{J}\left(W^{\{k\}}\right) - 2\mathcal{J}\left(W^{\{k-1\}}\right) + \mathcal{J}\left(W^{\{k-2\}}\right)}. \tag{2.11}$$

The approximation (2.11) may, however, lead to inaccurate estimates when the iteration and numerical errors are of comparable size.

2.1.4. Numerical example. We construct a diagonally dominant, symmetric 20×20 matrix $\mathbf{A} = 20\mathbf{I} + \mathbf{R}$ where $\mathbf{R}$ is a random 20×20 matrix with Frobenius norm less than or equal to one. We decompose $\mathbf{A}$ into two matrices $\mathbf{D}$ and $\mathbf{C}$, $\mathbf{A} = \mathbf{D} + \mathbf{C}$, where $\mathbf{D}$ contains only the diagonal entries of $\mathbf{A}$. We then solve $\mathbf{A}u = b$ via operator decomposition iteration with the iteration given by (2.1) where the solution at each iteration is obtained using conjugate gradient solver with a tolerance of 10^{-4}. The iteration terminates when $\|U^{\{i\}} - U^{\{i-1\}}\| < 10^{-8}$ which is accomplished in 11 iterations. We then solve (to within round-off) the sequence of 11 adjoint problems defined recursively by (2.4) and (2.5) for quantity of interest $w = u_{20}$, i.e., with $\mathbf{D}\phi^{[1]} = \psi = \mathbf{e}_{20}$ (where $\mathbf{e}_{20}$ is the unit vector with a single non-zero entry in the 20th row)and compute each of the terms

DOI: 10.1002/nme

Total Error $\lvert w_{20} - W^{\{11\}} \rvert$	Iteration Error $\lvert w - w^{\{11\}} \rvert$	Numerical Error $\lvert w^{\{11\}} - W^{\{11\}} \rvert$	Primary Numerical Error $\lvert \mathcal{R}(W^{\{11\}}, \phi^{[1]}; W^{\{10\}}) \rvert$
7.43823×10^{-5}	2.2234×10^{-11}	7.4382×10^{-5}	9.465×10^{-5}

Table 2.1. Error components for $W^{\{11\}}$ for example 2.1.4.

in the error representation formula (2.3) for $k = 1, \ldots, 11$. We call the first term in this sum the primary numerical error.

The expected decay of the contributions to the error $e_{20}^{\{11\}}$ that are inherited from previous iterations is illustrated in Figure 2.1.

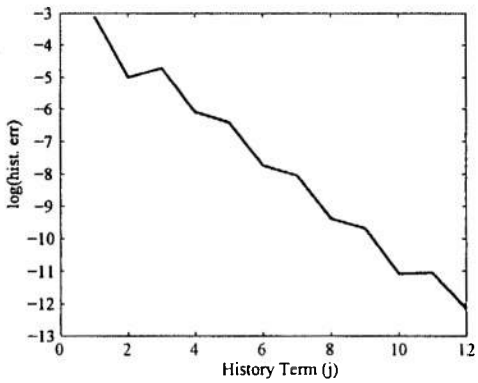


Figure 2.1. Individual "history" terms $\mathcal{R}(W^{\{11-j+1\}}, \phi^{[j]}; W^{\{11-j\}})$ for example 2.1.4 illustrating the expected decay in contributions to $e_{20}^{\{11\}}$ with index j.

2.2. *Estimating the numerical error for nonlinear algebraic systems*

Nonlinearity introduces additional complexities for defining an adjoint problem. We solve the nonlinear equation

$$\mathbf{A}w = f(w) \tag{2.12}$$

by successive approximation

$$\mathbf{A}w^{\{i\}} = f\left(w^{\{i-1\}}\right),\ i = 1, \ldots, k \tag{2.13}$$

with $W^{\{0\}} = 0$. Given $W^{\{0\}} = 0$, we compute the sequence of numerical approximations $W^{\{i\}}$ for $i = 1, \ldots, k$ as

$$\mathbf{A}W^{\{i\}} = f(W^{\{i-1\}}). \tag{2.14}$$

Theorem 2.2
The error in a quantity of interest is estimated as,

$$\left(e^{\{k\}}, \psi\right) = \sum_{j=1}^{k} \mathcal{R}\left(W^{\{k-j+1\}}, \phi^{[j]}; W^{\{k-j\}}\right), \tag{2.15}$$

where the adjoint problems are defined,

$$\left.\begin{aligned} &\mathbf{A}^{\mathsf{T}}\phi^{[1]} = \psi, \\ &\mathbf{A}^{\mathsf{T}}\phi^{[j]} = Lf(w^{\{k-j\}}, W^{\{k-j\}})^{\mathsf{T}}\phi^{[j-1]}, \quad j = 2, \ldots, k \end{aligned}\right\}, \tag{2.16}$$

DOI: 10.1002/nme

using the linearization $Lf(v,V)$ defined,

$$Lf(v,V) = \int_0^1 \mathbf{J}(sv + (1-s)V)\, ds,$$

where $\mathbf{J}(\cdot)$ is the Jacobian of f.

Proof
The steps are similar to the previous arguments,

$$\begin{aligned}
\left(e^{\{k\}}, \psi\right) &= \left(e^{\{k\}}, \mathbf{A}^\top \phi^{[1]}\right) \\
&= \left(\mathbf{A}w^{\{k\}} - \mathbf{A}W^{\{k\}}, \phi^{[1]}\right) \\
&= \left(f(w^{\{k-1\}}) - \mathbf{A}W^{\{k\}}, \phi^{[1]}\right) \\
&= \left(f(W^{\{k-1\}}) - \mathbf{A}W^{\{k\}}, \phi^{[1]}\right) + \left(f(w^{\{k-1\}}) - f(W^{\{k-1\}}), \phi^{[1]}\right) \\
&= \mathcal{R}\left(W^{\{k\}}, \phi^{[1]}; W^{\{k-1\}}\right) + \left(Lf(w^{\{k-1\}}, W^{\{k-1\}})e^{\{k-1\}}, \phi^{[1]}\right) \\
&= \mathcal{R}\left(W^{\{k\}}, \phi^{[1]}; W^{\{k-1\}}\right) + \left(e^{\{k-1\}}, Lf(w^{\{k-1\}}, W^{\{k-1\}})^\top \phi^{[1]}\right) \\
&= \mathcal{R}\left(W^{\{k\}}, \phi^{[1]}; W^{\{k-1\}}\right) + \left(e^{\{k-1\}}, \mathbf{A}^\top \phi^{[2]}\right) \\
&\vdots \\
&= \sum_{j=1}^{k} \mathcal{R}\left(W^{\{k-j+1\}}, \phi^{[j]}; W^{\{k-j\}}\right).
\end{aligned}$$

□

2.2.1. Linearization and adjoints for nonlinear adjoints. In general, there are multiple ways to define an adjoint to a nonlinear operator [10]. Choosing a suitable definition is highly dependent on the purpose for which the adjoint is intended. For a perturbation or error analysis, one systematic way to define an adjoint is based on linearization. Consider an approximate solution $W \approx w$ of (2.12) computed without iteration, and define the residual

$$\mathcal{R}(W) = \mathbf{A}W - f(W).$$

The standard analysis begins

$$\mathbf{A}(w - W) - (f(w) - f(W)) = -\mathcal{R}(W).$$

The integral Mean Value Theorem yields

$$f(w) - f(W) = Lf\,(w - W) = \int_0^1 \mathbf{J}(sw + (1-s)W)\, ds\, (w - W).$$

Introducing the formal adjoint problem

$$(\mathbf{A} - Lf)^* \phi = \psi$$

leads to the *a posteriori* error estimate

$$(w - W, \psi) = (-\mathcal{R}(W), \phi).$$

In practice, the linearization Lf cannot be computed and it is approximated as,

$$Lf(v,V) = \int_0^1 \mathbf{J}(sv + (1-s)V)\, ds \approx \int_0^1 \mathbf{J}(sV + (1-s)V)\, ds = \mathbf{J}(V). \tag{2.17}$$

DOI: 10.1002/nme

The approximation in the linearization may certainly affect the accuracy of the *a posteriori* error estimate. However, the effect can be bounded *a priori* under regularity assumptions on the problem, i.e. the second derivatives of f are uniformly bounded in a compact region containing the true solution, analytic iterative solutions, and iterative numerical solutions, and assuming that the iteration converges and the approximations are sufficiently close to the true solution. In particular W should be sufficiently close to w and $\mathcal{R}(W)$ should be sufficiently small. In practice, the approximation (2.17) works well in many situations in the sense that the linearization error has relatively insignificant effect on accuracy of the estimate when the numerical solutions are reasonably accurate. Perhaps more importantly, catastrophic failure of the *a posteriori* error estimate, that is an estimate that is low when the actual error is large, is relatively difficult to manage. See [8] for a discussion of this point for systems of reaction-diffusion equations and an example where catastrophic failure is created.

However, this simple approach to defining an adjoint operator when operator decomposition iteration is used can fail. The difficulty is that an iterative solution $W^{\{i\}}$ is actually solving a problem that is significantly different than the original problem. See [2] for a discussion of this point.

The approach to define an adjoint for Theorem 2.2 avoids this issue by constructing a global adjoint for the iterative solution operator via a sequence of coupled adjoints for each individual component problem obtained by linearization between the iterative analytic $w^{\{i\}}$ and iterative numeric $W^{\{i\}}$ solutions. The effects of this "local" linearization can be controlled under local regularity and convergence assumptions as described above. In practice,

$$\mathbf{A}^{\mathsf{T}}\phi^{[j]} = Lf(w^{\{k-j\}}, W^{\{k-j\}})^{\mathsf{T}}\phi^{[j-1]}$$

is approximated by

$$\mathbf{A}^{\mathsf{T}}\tilde{\phi}^{[j]} = \mathbf{J}(W^{\{k-j\}})^{\mathsf{T}}\tilde{\phi}^{[j-1]} \tag{2.18}$$

for $j = 2, \ldots, k$.

The price of the indicated approach to defining an adjoint for the iterated solution operator is that the *a posteriori* error estimates accounts only for the numerical error $e^{\{i\}}$ and leaves the analytic iterative error $\mathcal{E}^{\{i\}}$ remaining to be estimated. We adapt the estimate discussed in Sec. 2.1.3.

2.2.2. Numerical examples. We now consider three new situations that may arise for nonlinearly coupled problems. Let

$$\mathbf{A}w = f(w),$$

where

$$\mathbf{A} = \begin{pmatrix} 3 & -1 & 0 & 0 \\ -1 & 3 & 0 & 0 \\ 0 & 0 & 3 & -1 \\ 0 & 0 & -1 & 3 \end{pmatrix} \quad \text{and} \quad f(w) = \alpha \begin{pmatrix} \exp(-\beta(w_3 - 0.4)^2) \\ \exp(-\beta(w_4 - 0.6)^2) \\ \exp(-\beta(w_1 - 0.5)^2) \\ \exp(-\beta(w_2 - 0.3)^2) \end{pmatrix},$$

and let the quantity of interest be the value of w_2. In all cases, a high accuracy reference solution w was generated using Newton's method with a tolerance of 10^{-12} while we also computed high accuracy reference iterative solutions $w^{\{i\}}$ to full precision. To compare to estimates employing the adjoint problem associated with the original system, the "global" adjoint corresponding to the full (i.e. non-decomposed) problem was also constructed and solved and error estimates based on the adjoint to the full problem are reported in the following examples. We set,

- $\mathcal{R}(W, \phi)$ is the error estimate obtianed by solving an adjoint problem using the exact linearization of the global adjoint problem, *i.e.* linearizing around the average of the reference solution w and the approximate solution W, while
- $\mathcal{R}(W, \tilde{\phi})$ is the error estimate obtained from using an approximate linearization of the global adjoint problem.

The approximate solutions $W^{\{i\}}$ were computed by rounding to the 6th decimal place. All adjoint solutions were computed with full precision. In each case we estimated the iteration error using (2.11).

DOI: 10.1002/nme

Increasing the value of α essentially reduces the diagonal dominance of the operator decomposition, damaging iterative convergence, while increasing β raises the linearization error.

Cancellation between iteration and numerical error. Let $\alpha = 1$, $\beta = 10$, $i = 40$. The iteration converges after 40 iterations when the norm of the residual is less than 10^{-4}. Results are provided in Table 2.2.

Error $\lvert w_2 - W_2 \rvert$	Estimated Error $\mathcal{R}(W, \phi)$	Practical Error $\mathcal{R}(W, \tilde{\phi})$	Iteration Error $w_2 - w_2^{\{k\}}$
1.1102276×10^{-6}	1.1102276×10^{-6}	1.10983×10^{-6}	-2.55547×10^{-6}
Numerical Error $w_2^{\{k\}} - W_2^{\{k\}}$	**Est. Numerical Error** $\sum_j \mathcal{R}(W^{\{k-j+1\}}, \phi^{[j]})$	**Pract. Numerical Error** $\sum_j \mathcal{R}(W^{\{k-j+1\}}, \tilde{\phi}^{[j]})$	**Est. Iteration Error**
1.44524×10^{-6}	1.44524×10^{-6}	1.44592×10^{-6}	-6.6554×10^{-6}

Table 2.2. Error components for example 2.2.2. Note the cancellation of errors between the iteration and numerical error. ($\mathcal{R}(W^{\{k-j+1\}}, \phi^{[j]}; W^{\{k-j\}}) \equiv \mathcal{R}(W^{\{k-j+1\}}, \phi^{[j]})$.)

Notice that the adjoint to the coupled problem gives an excellent estimate of the error, yielding four significant figures even when using the approximate linearization. What is obscured is the cancellation that occurs between the iteration and numerical error. The methods developed here enable the iteration and numerical errors to be estimated separately and this is important when constructing adaptive algorithms. However, the iteration error estimate is polluted by the numerical error in $U^{\{k\}}$. The partial sums of the history terms are plotted in Figure 2.2 where expected decay of history error contributions can be observed.

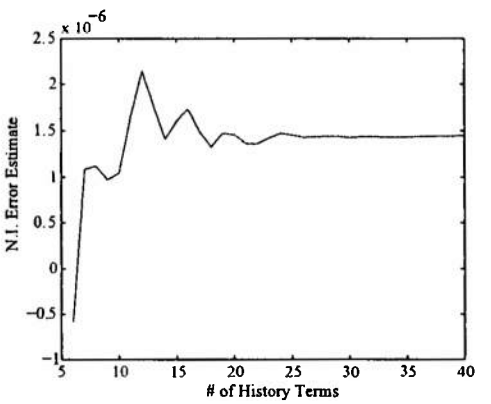


Figure 2.2. Numerical error estimate $\sum_{j=1}^{m} \mathcal{R}(U^{\{k-j+1\}}, \phi^{[j]}; U^{\{k-j\}})$ including m "history" terms for Ex. 2.2.2.

The effect of linearization error. Let $\alpha = 2$, $\beta = 8$, $i = 100$. In this example, the iteration fails to converge. In addition, $w^{\{k\}}$ and $W^{\{k\}}$ approach they same fixed point, but hey both do so in a non-monotonic fashion. This produces significant differences between $Lf(w^{\{k-j\}}, W^{\{k-j\}})^{\mathsf{T}}$ and $\mathbf{J}(W^{\{k-j\}})^{\mathsf{T}}$ (see equations (2.16) and (2.18) respectively) and consequently significant differences in the corresponding adjoint solutions ϕ and $\tilde{\phi}$. Results are provided in Table 2.4 where the practical numerical error estimate is seen to be completely inaccurate.

DOI: 10.1002/nme

To obtain an estimate of how rapidly the adjoint solutions ϕ and $\tilde{\phi}$ can diverge, we see from (2.16) and (2.18) that

$$\begin{aligned}\phi^{[j]} - \tilde{\phi}^{[j]} &= \mathbf{A}^{-\mathsf{T}}\left(Lf(W^{\{k-j\}}, w^{\{k-j\}})^{\mathsf{T}}\phi^{[j-1]} - \mathbf{J}(W^{\{k-j\}})^{\mathsf{T}}\tilde{\phi}^{[j-1]}\right)\\ &\approx \mathbf{A}^{-\mathsf{T}}\left(Lf(W^{\{k-j\}}, w^{\{k-j\}}) - \mathbf{J}(W^{\{k-j\}})\right)^{\mathsf{T}}\phi^{[j-1]}\end{aligned}$$

so the spectral radius of $\mathbf{A}^{-\mathsf{T}}\left(Lf(W^{\{k-j\}}, w^{\{k-j\}}) - \mathbf{J}(W^{\{k-j\}})\right)^{\mathsf{T}}$ can be considered as an amplification factor to explain the exponential accumulation of linearization error in the history error estimate.

Error $w_2 - W_2$	Estimated Error $\mathcal{R}(W, \phi)$	Practical Error $\mathcal{R}(W, \tilde{\phi})$	Iteration Error $w_2 - w_2^{\{k\}}$
-0.048166	0.048166	7.06634	-0.050168
Numerical Error $w_2^{\{k\}} - W_2^{\{k\}}$	**Est. Numerical Error** $\sum_j \mathcal{R}(W^{\{j-k\}}, \phi^{[j]})$	**Pract. Numerical Error** $\sum_j \mathcal{R}(W^{\{j-k\}}, \tilde{\phi}^{[j]})$	**Est. Iteration Error**
0.002002	0.002002	3.58×10^6	-0.04568

Table 2.3. Error components for Ex. 2.2.2, with $\mathcal{R}(W^{\{k-j+1\}}, \phi^{[j]}) \equiv \mathcal{R}(W^{\{k-j+1\}}, \phi^{[j]}; W^{\{k-j\}})$.

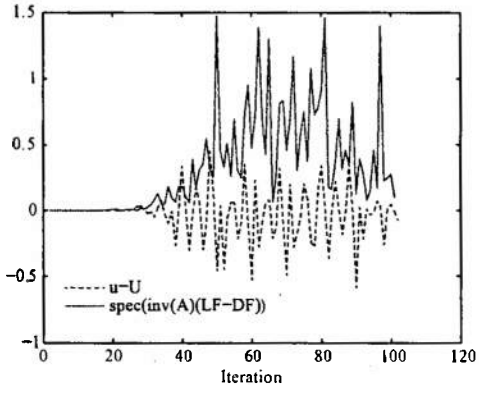


Figure 2.3. Differences between $w_2^{\{k\}}$ and $W_2^{\{k\}}$ and the spectral radius of $\mathbf{A}^{-\mathsf{T}}(Lf(w, W) - J(W))^{\mathsf{T}}$ for Ex. 2.2.2.

Divergent iteration Let $\alpha = 2$, $\beta = 7$, $k = 100$. The iteration fails to converge, but all error estimates in Table 2.4 are well-behaved but exhibit significant linearization error. Not surprisingly since the iteration error estimate assumes a convergent iteration, the estimate of the iteration error is poor. However, despite the fact that the iteration is divergent, the estimate of the numerical error is accurate and the effectivity ratio (defined as the ratio of the error estimate to the true error) shown in Figure 2.4 improves as the number of history terms is increased, although the practical numerical error estimate is affected by errors resulting from the linearization.

DOI: 10.1002/nme

Error $\lvert w_2 - W_2 \rvert$	Estimated Error $\mathcal{R}(W, \phi)$	Practical Error $\mathcal{R}(W, \tilde{\phi})$	Iteration Error $w_2 - w_2^{\{k\}}$
0.0999965	0.0999965	0.076506	0.0999963
Numerical Error $w_2^{\{k\}} - W_2^{\{k\}}$	**Est. Numerical Error** $\sum_j \mathcal{R}(W^{\{j-k\}}, \phi^{[j]})$	**Pract. Numerical Error** $\sum_j \mathcal{R}(W^{\{j-k\}}, \tilde{\phi}^{[j]})$	**Est. Iteration Error**
1.8524×10^{-6}	1.8524×10^{-6}	1.8530×10^{-6}	0.18273

Table 2.4. Error components for Ex. 2.2.2. ($\mathcal{R}(W^{\{k-j+1\}}, \phi^{[j]}) \equiv \mathcal{R}(W^{\{k-j+1\}}, \phi^{[j]}; W^{\{k-j\}})$.

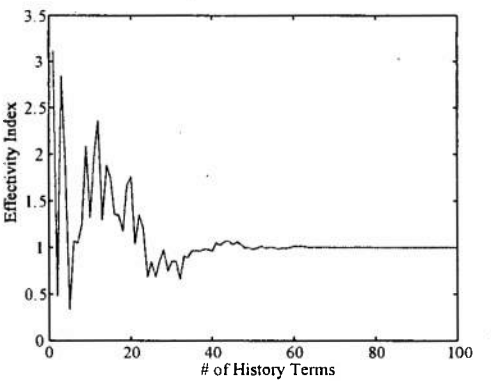


Figure 2.4. Effectivity ratio for the numerical error estimate $\sum_{j=1}^{m} \mathcal{R}(W^{\{k-j+1\}}, \phi^{[j]}; W^{\{k-j\}})$ including m "history" terms for Ex. 2.2.2.

2.2.3. Discussion. An interesting case is the situation in which the numerical error is much higher than the stopping criteria (say round U to the third decimal place), but the corresponding theoretical iteration converges quickly. The computation does not converge due to the numerical error, but the quick computation of a few history terms leads to an exact estimate of the numerical error. See Section 4.2.4 for an adaptive solution to a similar problem.

In all three examples, the global adjoints are not diagonally dominant (or SPD). But even when the operator decomposition iteration solution for the global adjoint (*e.g.* a "Jacobi" iteration) does not converge, the *a posteriori error* analysis still provides meaningful error estimates. See Section 4.2.2 for more discussion.

3. ANALYSIS FOR SEMILINEAR ELLIPTIC SYSTEMS

For ease of presentation, we focus the analysis on a two component fully coupled elliptic system of the form

$$\begin{aligned} -\nabla \cdot a_1(x)\nabla u_1 + b_1(x) \cdot \nabla u_1 + c_1(x)u_1 &= f_1(x, u_1, u_2, Du_2), \quad x \in \Omega, \\ -\nabla \cdot a_2(x)\nabla u_2 + b_2(x) \cdot \nabla u_2 + c_2(x)u_2 &= f_2(x, u_1, Du_1, u_2), \quad x \in \Omega, \\ u_1 = u_2 &= 0, \quad x \in \partial\Omega, \end{aligned} \tag{3.1}$$

where Ω is a convex polygonal domain in $\mathbb{R}^i, i = 1, 2, 3$, with boundary $\partial\Omega$, and we assume that $a_i, b_i, c_i, f_i, i = 1, 2$ are sufficiently smooth to establish optimal order *a priori* convergence for the finite element method computed without operator decomposition iteration.

DOI: 10.1002/nme

The weak formulation of (3.1) is: find $u_m \in \tilde{W}_2^1(\Omega)$ satisfying

$$\begin{aligned} \mathcal{A}_1(u_1, v_1) &= (f_1(x, u_1, u_2, Du_2), v_1), \quad \forall v_1 \in \tilde{W}_2^1(\Omega), \\ \mathcal{A}_2(u_2, v_2) &= (f_2(x, u_1, Du_1, u_2), v_2), \quad \forall v_2 \in \tilde{W}_2^1(\Omega), \end{aligned} \tag{3.2}$$

where

$$\begin{aligned} \mathcal{A}_1(u_1, v_1) &\equiv (a_1 \nabla u_1, \nabla v_1) + (b_1 \cdot \nabla u_1, v_1) + (c_1 u_1, v_1), \\ \mathcal{A}_2(u_2, v_2) &\equiv (a_2 \nabla u_2, \nabla v_2) + (b_2 \cdot \nabla u_2, v_2) + (c_2 u_2, v_2), \end{aligned}$$

are assumed to be coercive bilinear forms on Ω and $\tilde{W}_p^n(\Omega)$ represents the subspace of $W_p^n(\Omega)$ with zero trace on $\partial\Omega$.

If we were numerically solving (3.2) without operator decomposition iteration, we would introduce conforming discretizations $\mathcal{S}_{h,m}(\Omega)$ and then solve the discretized system: find $U_m \in \mathcal{S}_{h,m}(\Omega)$ satisfying

$$\begin{aligned} \mathcal{A}_1(U_1, \chi_1) &= (f_1(x, U_1, U_2, DU_2), \chi_1), \quad \forall \chi_1 \in \mathcal{S}_{h,m}(\Omega), \\ \mathcal{A}_2(U_2, \chi_2) &= (f_2(x, U_1, DU_1, U_2), \chi_2), \quad \forall \chi_2 \in \mathcal{S}_{h,m}(\Omega). \end{aligned} \tag{3.3}$$

3.1. *Analysis of operator decomposition iteration solutions*

We analyze three different operator decomposition iterations for producing numerical approximations of (3.1). We recall that we decompose the error as in (1.2), i.e.,

$$u - U^{\{k\}} = \underbrace{u - u^{\{k\}}}_{\text{analytic iteration error}} + \underbrace{u^{\{k\}} - U^{\{k\}}}_{\text{numerical error}} = \mathcal{E}^{\{k\}} + e^{\{k\}},$$

where $u^{\{k\}}$ respectively $U^{\{k\}}$ are the analytic and discrete solutions obtained by the particular iteration approach under consideration. The *a posteriori* error analysis is for the numerical error $e^{\{k\}}$. We estimate the analytic iteration error $\mathcal{E}^{\{k\}}$ by a natural application of the estimates discussed in Sec. 2.1.3. To simplify the analysis, if the resulting operator decomposition elliptic equation for U^i is nonlinear (in U^i), we assume that the error introduced by its nonlinear solve is negligible.

For simplicity of presentation, we assume the quantity of interest is given as a linear functional of the second solution component, determined by $\psi = (0, \psi_2)^T$, i.e. we estimate $(e_2^{\{k\}}, \psi_2)$. For notational convenience and to be consistent with [1], we abbreviate the *weak residual* of a solution component,

$$\mathcal{R}_m(\mu, \chi; \nu) = (f_m(\nu), \chi) - \mathcal{A}_m(\mu, \chi), \quad m = 1, 2.$$

3.1.1. Block Jacobi iteration We first consider a block Jacobi operator decomposition iterative approach to the solution of the semilinear elliptic system (3.1).

Algorithm 1 Block Jacobi algorithm

Given $U_1^{\{0\}}$ and $U_2^{\{0\}}$
for $i = 1, 2, 3, \ldots$ until convergence **do**
  Find $U_1^{\{i\}} \in \mathcal{S}_{h,1}(\Omega) \ni \mathcal{A}_1(U_1^{\{i\}}, \chi_1) = (f_1(x, U_1^{\{i\}}, U_2^{\{i-1\}}, DU_2^{\{i-1\}}), \chi_1) \; \forall \chi_1 \in \mathcal{S}_{h,1}(\Omega)$
  Find $U_2^{\{i\}} \in \mathcal{S}_{h,2}(\Omega) \ni \mathcal{A}_2(U_2^{\{i\}}, \chi_2) = (f_2(x, U_1^{\{i-1\}}, DU_1^{\{i-1\}}, U_2^{\{i\}}), \chi_2) \; \forall \chi_2 \in \mathcal{S}_{h,2}(\Omega)$
end for

Theorem 3.1
The representation formula for the numerical error $e^{\{k\}}$ is

$$(e^{\{k\}}, \psi) = \sum_{j \le k,\ j \text{ odd}} \mathcal{R}_2(U_2^{\{k-j+1\}}, \phi_2^{[j]}; U_1^{\{k-j\}}) + \sum_{j \le k,\ j \text{ even}} \mathcal{R}_1(U_1^{\{k-j+1\}}, \phi_1^{[j]}; U_2^{\{k-j\}}), \tag{3.4}$$

DOI: 10.1002/nme

where the corresponding adjoint problems are

$$\begin{aligned} \mathcal{A}_2^*(\chi, \phi_2^{[j]}) &= (\chi, \psi_2^{[j]}), \quad \forall \chi \in \tilde{W}_2^1(\Omega), \quad j \le k,\ j \text{ odd}, \\ \mathcal{A}_1^*(\chi, \phi_1^{[j]}) &= (\chi, \psi_1^{[j]}), \quad \forall \chi \in \tilde{W}_2^1(\Omega), \quad j \le k,\ j \text{ even}, \end{aligned} \tag{3.5}$$

with

$$\begin{aligned} \mathcal{A}_1^*(v, w) &= (\nabla v, a_1 \nabla w) - (v, \operatorname{div}(b_1 w)) + (v, (c_1 - J_{1,1}(U))w), \\ \mathcal{A}_2^*(v, w) &= (\nabla v, a_2 \nabla w) - (v, \operatorname{div}(b_2 w)) + (v, (c_2 - J_{2,2}(U))w), \end{aligned} \tag{3.6}$$

and the additional adjoint data is defined recursively as

$$\begin{aligned} \psi_2^{[1]} &= \psi_1, \psi_1^{[1]} = 0, \\ (\chi, \psi_2^{[j]}) &= (J_{1,2}(U^{\{k-j+1\}})\chi, \phi_1^{[j-1]}), \quad j \le k,\ j \text{ odd}, \\ (\chi, \psi_1^{[j]}) &= (J_{2,1}(U^{\{k-j+1\}})\chi, \phi_2^{[j-1]}), \quad j \le k,\ j \text{ even}, \end{aligned} \tag{3.7}$$

where $J_{m,n}(V)$ is the Jacobian of f_m with respect to u_n evaluated at V.

If we wish to highlight the final (kth) iteration, then we can write

$$\begin{aligned} (e^{\{k\}}, \psi) = \mathcal{R}_2(U_2^{\{k\}}, \phi_2^{[1]}; U_1^{\{k-1\}}) &+ \sum_{2 \le j \le k,\ j \text{ even}} \mathcal{R}_1(U_1^{\{k-j+1\}}, \phi_1^{[j]}; U_2^{\{k-j\}}) \\ &+ \sum_{3 \le j \le k,\ j \text{ odd}} \mathcal{R}_2(U_2^{\{k-j+1\}}, \phi_2^{[j]}; U_1^{\{k-j\}}). \end{aligned} \tag{3.8}$$

Proof

In the following, we simplify notation in the functions f, so for example we write $f_1(x, u_1, u_2, Du_2) \equiv f_1(u_2)$, and so on.

$$\begin{aligned} (e^{\{k\}}, \psi) &= (e_2^{\{k\}}, \psi_2^{[1]}) \\ &= \mathcal{A}_2^*(e_2^{\{k\}}, \phi_2^{[1]}) \\ &= \mathcal{A}_2(e_2^{\{k\}}, \phi_2^{[1]}) \\ &= \mathcal{A}_2(u_2^{\{k\}}, \phi_2^{[1]}) - \mathcal{A}_2(U_2^{\{k\}}, \phi_2^{[1]}) \\ &= (f_2(u_1^{\{k-1\}}), \phi_2^{[1]}) - \mathcal{A}_2(U_2^{\{k\}}, \phi_2^{[1]}) \\ &= (f_2(U_1^{\{k-1\}}), \phi_2^{[1]}) - \mathcal{A}_2(U_2^{\{k\}}, \phi_2^{[1]}) + (f_2(u_1^{\{k-1\}}), \phi_2^{[1]}) - (f_2(U_1^{\{k-1\}}), \phi_2^{[1]}) \\ &= \mathcal{R}_2(U_2^{\{k\}}, \phi_2^{[1]}; U_1^{\{k-1\}}) + (Lf_2(u_1^{\{k-1\}}, U_1^{\{k-1\}})e_1^{\{k-1\}}, \phi_2^{[1[}) \\ &\approx \mathcal{R}_2(U_2^{\{k\}}, \phi_2^{[1]}; U_1^{\{k-1\}}) + (J_2(U_1^{\{k-1\}})e_1^{\{k-1\}}, \phi_2^{[1]}) \\ &= \mathcal{R}_2(U_2^{\{k\}}, \phi_2^{[1]}; U_1^{\{k-1\}}) + (e_1^{\{k-1\}}, \psi_1^{[2]}) \\ &= \mathcal{R}_2(U_2^{\{k\}}, \phi_2^{[1]}; U_1^{\{k-1\}}) + \mathcal{A}_1^*(e_1^{\{k-1\}}, \phi_1^{[2]}) \\ &= \mathcal{R}_2(U_2^{\{k\}}, \phi_2^{[1]}; U_1^{\{k-1\}}) + \mathcal{A}_1(e_1^{\{k-1\}}, \phi_1^{[2]}) \\ &= \mathcal{R}_2(U_2^{\{k\}}, \phi_2^{[1]}; U_1^{\{k-1\}}) + \mathcal{R}_1(U_1^{\{k-1\}}, \phi_1^{[2]}; U_2^{\{k-2\}}) \\ &\qquad + (Lf_1(u_2^{\{k-2\}}, U_2^{\{k-2\}})e_2^{\{k-2\}}, \phi_1^{[2]}) \\ &\approx \mathcal{R}_2(U_2^{\{k\}}, \phi_2^{[1]}; U_1^{\{k-1\}}) + \mathcal{R}_2(U_1^{\{k-1\}}, \phi_1^{[2]}; U_2^{\{k-2\}}) + (J_1(U_2^{\{k-2\}})e_2^{\{k-2\}}, \phi_1^{[2]}) \\ &= \mathcal{R}_2(U_2^{\{k\}}, \phi_2^{[1]}; U_1^{\{k-1\}}) + \mathcal{R}_1(U_1^{\{k-1\}}, \phi_1^{[2]}; U_2^{\{k-2\}}) + \mathcal{A}_2^*(e_2^{\{k-2\}}, \psi_2^{[3]}) \\ &\ \ \vdots \\ &= \sum_{j \le k,\ j \text{ odd}} \mathcal{R}_2(U_2^{\{k-j+1\}}, \phi_2^{[j]}; U_1^{\{k-j\}}) + \sum_{j \le k,\ j \text{ even}} \mathcal{R}_1(U_1^{\{k-j+1\}}, \phi_1^{[j]}; U_2^{\{k-j\}}). \end{aligned} \tag{3.9}$$

DOI: 10.1002/nme

□

The estimate in this case takes into account the numerical error arising from each component solution and the inherited errors passed between iterations. Following the discussion in Sec. 2.2.1, we are using the adjoint naturally associated with the analytic iterative solution operator. We use "local" linearization $Lf_i(u_j^{\{k\}}, U_j^{\{k\}})$ between $u_j^{\{k\}}$ and $U_j^{\{k\}}$ to define the required adjoints for each component equation in the iteration. The effects of this linearization can be controlled assuming sufficient regularity of the solution and *a priori* convergence of the discretization, and we expect the estimates to be accurate for all sufficiently accurate numerical solutions. The global adjoint of the iterated solution operator is therefore obtained by a sequence of "single physics solves". We note that we still have to estimate the analytic iteration error $\mathcal{E}^{\{k\}}$.

3.1.2. Block Gauss-Seidel iteration Next, we consider a block Gauss-Seidel operator decomposition iteration.

Algorithm 2 Block Gauss-Seidel algorithm

Given $U_2^{\{0\}}$,
for $i = 1, 2, 3, \ldots$ until convergence **do**
  Find $U_1^{\{i\}} \in \mathcal{S}_{h,1}(\Omega) \ni \mathcal{A}_1(U_1^{\{i\}}, \chi_1) = \big(f_1(x, U_1^{\{i\}}, U_2^{\{i-1\}}, DU_2^{\{i-1\}}), \chi_1\big) \forall \chi_1 \in \mathcal{S}_{h,1}(\Omega)$
  Find $U_2^{\{i\}} \in \mathcal{S}_{h,2}(\Omega) \ni \mathcal{A}_2(U_2^{\{i\}}, \chi_2) = \big(f_2(x, U_1^{\{i\}}, DU_1^{\{i\}}, U_2^{\{i\}}), \chi_2\big) \forall \chi_2 \in \mathcal{S}_{h,2}(\Omega)$
end for

Theorem 3.2
The representation formula for the numerical error $e^{\{k\}}$ is

$$\big(e^{\{k\}}, \psi\big) = \sum_{j=1}^{k} \underbrace{\mathcal{R}_2\big(U_2^{\{k-j+1\}}, \phi_2^{[j]}; U_1^{\{k-j+1\}}\big)}_{\text{within iteration errors}} + \sum_{j=1}^{k} \underbrace{\mathcal{R}_1\big(U_1^{\{k-j+1\}}, \phi_1^{[j]}; U_2^{\{k-j\}}\big)}_{\text{between iteration errors}}, \tag{3.10}$$

where the corresponding adjoint problems are

$$\begin{aligned} \mathcal{A}_2^*\big(\chi, \phi_2^{[j]}\big) &= \big(\chi, \psi_2^{[j]}\big), \quad \forall \chi \in \tilde{W}_2^1(\Omega), \quad j = 1, \ldots, k, \\ \mathcal{A}_1^*\big(\chi, \phi_1^{[j]}\big) &= \big(\chi, \psi_1^{[j]}\big), \quad \forall \chi \in \tilde{W}_2^1(\Omega), \quad j = 1, \ldots, k, \end{aligned} \tag{3.11}$$

and the adjoint data is defined recursively as

$$\begin{aligned} \psi_2^{[1]} &= \psi, \psi_1^{[1]} = 0, \\ \big(\chi, \psi_2^{[j]}\big) &= \big(J_1(U^{\{k-j+1\}})\chi, \phi_1^{[j-1]}\big), \quad j = 2, \ldots, k, \\ \big(\chi, \psi_1^{[j]}\big) &= \big(J_2(U^{\{k-j+1\}})\chi, \phi_2^{[j]}\big), \quad j = 1, \ldots, k. \end{aligned} \tag{3.12}$$

Proof

DOI: 10.1002/nme

$$
\begin{aligned}
\big(e^{\{k\}},\psi\big) &= \big(e_2^{\{k\}},\psi_2^{[1]}\big)\\
&= \big(f_2(u_1^{\{k\}}),\phi_2^{[1]}\big) - \mathcal{A}_2\big(U_2^{\{k\}},\phi_2^{[1]}\big)\\
&= \mathcal{R}_2\big(U_2^{\{k\}},\phi_2^{[1]};U_1^{\{k\}}\big) + \big(Lf_2(u_1^{\{k\}},U_1^{\{k\}})e_1^{\{k\}},\phi_2^{[1]}\big)\\
&\approx \mathcal{R}_2\big(U_2^{\{k\}},\phi_2^{[1]};U_1^{\{k\}}\big) + \big(J_2(U_1^{\{k\}})e_1^{\{k\}},\phi_2^{[1]}\big)\\
&= \mathcal{R}_2\big(U_2^{\{k\}},\phi_2^{[1]};U_1^{\{k\}}\big) + \big(e_1^{\{k\}},\psi_1^{[1]}\big)\\
&= \mathcal{R}_2\big(U_2^{\{k\}},\phi_2^{[1]};U_1^{\{k\}}\big) + \big(f_1(U_2^{\{k-1\}}),\phi_1^{[1]}\big) - \mathcal{A}_1\big(U_1^{\{k\}},\phi_1^{[1]}\big)\\
&\qquad +\big(f_1(u_2^{\{k-1\}}),\phi_1^{[1]}\big) - \big(f_1(U_2^{\{k-1\}}),\phi_1^{[1]}\big)\\
&= \mathcal{R}_2\big(U_2^{\{k\}},\phi_2^{[1]};U_1^{\{k\}}\big) + \mathcal{R}_1\big(U_1^{\{k\}},\phi_1^{[1]};U_2^{\{k-1\}}\big)\\
&\qquad\qquad +\big(Lf_1(u_2^{\{k-1\}},U_2^{\{k-1\}})e_2^{\{k-1\}},\phi_1^{[1]}\big)\\
&\approx \mathcal{R}_2\big(U_2^{\{k\}},\phi_2^{[1]};U_1^{\{k\}}\big) + \mathcal{R}_2\big(U_1^{\{k\}},\phi_1^{[1]};U_2^{\{k-1\}}\big) + \big(J_1(U_2^{\{k-1\}})e_2^{\{k-1\}},\phi_1^{[1]}\big)\\
&= \mathcal{R}_2\big(U_2^{\{k\}},\phi_2^{[1]};U_1^{\{k\}}\big) + \mathcal{R}_1\big(U_1^{\{k\}},\phi_1^{[1]};U_2^{\{k-1\}}\big) + \mathcal{A}_2^*\big(e_2^{\{k-1\}},\psi_2^{[2]}\big)\\
&\ \ \vdots\\
&= \sum_{j=1}^{k}\mathcal{R}_2\big(U_2^{\{k-j+1\}},\phi_2^{[j]};U_1^{\{k-j+1\}}\big) + \sum_{j=1}^{k}\mathcal{R}_1\big(U_1^{\{k-j+1\}},\phi_1^{[j]};U_2^{\{k-j\}}\big). \qquad (3.13)
\end{aligned}
$$

□

In this case, in contrast with (3.4), there are contributions to the error reflecting both "within" and "between iteration" errors.

3.1.3. Relaxed block Jacobi iteration Both the Jacobi and Gauss-Seidel operator decomposition iterations can be "relaxed" by letting $U^{\{k\}} = \alpha\tilde{U}^{\{k\}} + (1-\alpha)U^{\{k-1\}}$, where a tilde denotes a quantity computed before relaxation. The approximation is obtained via

Algorithm 3 Relaxed block Jacobi algorithm

Given $U_1^{\{0\}}$ and $U_2^{\{0\}}$,
for $i = 1, 2, 3, \ldots$ until convergence **do**
  Find $\tilde{U}_1^{\{i\}} \in \mathcal{S}_{h,1}(\Omega) \ni \mathcal{A}_1\big(\tilde{U}_1^{\{i\}},\chi_1\big) = \big(f_1(x,\tilde{U}_1^{\{i\}},U_2^{\{i-1\}},DU_2^{\{i-1\}}),\chi_1\big)\ \forall\chi_1 \in \mathcal{S}_{h,1}(\Omega)$
  Calculate new iterate $U_1^{\{i\}} = \alpha\tilde{U}_1^{\{i\}} + (1-\alpha)U_1^{\{i-1\}}$
  Find $\tilde{U}_2^{\{i\}} \in \mathcal{S}_{h,2}(\Omega) \ni \mathcal{A}_2\big(\tilde{U}_2^{\{i\}},\chi_2\big) = \big(f_2(x,U_1^{\{i-1\}},DU_1^{\{i-1\}},\tilde{U}_2^{\{i\}}),\chi_2\big)\ \forall\chi_2 \in \mathcal{S}_{h,2}(\Omega)$
  Calculate new iterate $U_2^{\{i\}} = \alpha\tilde{U}_2^{\{i\}} + (1-\alpha)U_2^{\{i-1\}}$
end for

This is often done in practice in order to aid convergence of the iteration, but poses more challenges for *a posteriori* error analysis and we present a partial analysis to explain how the results can be extended to handle relaxation. Letting $\tilde{U}^{\{k\}}$, $\tilde{u}^{\{k\}}$, and $\tilde{e}^{\{k\}}$ represent the corresponding quantities computed without relaxation, we have

$$
\big(u - U^{\{k\}},\psi\big) = \big(u - u^{\{k\}},\psi\big) + \alpha\big(\tilde{e}^{\{k\}},\psi\big) + (1-\alpha)\big(e^{\{k-1\}},\psi\big). \qquad (3.14)
$$

DOI: 10.1002/nme

Using the notation above, we have

$$
\begin{aligned}
\left(e^{\{k\}},\psi\right) &= \left(e_2^{\{k\}},\psi_2^{[1]}\right) \\
&= \alpha\left(\tilde{e}_2^{\{k\}},\psi_2^{[1]}\right)+(1-\alpha)\left(e_2^{\{k-1\}},\psi_2^{[1]}\right) \\
&= \alpha\left[\mathcal{R}_2\left(\tilde{U}_2^{\{k\}},\phi_2^{[1]};U_1^{\{k-1\}}\right)+\left(f_2(u_1^{\{k-1\}}),\phi_2^{[1]}\right)-\left(f_2(U_1^{\{k-1\}}),\phi_2^{[1]}\right)\right] \\
&\qquad +(1-\alpha)\left(e_2^{\{k-1\}},\psi_2^{[1]}\right) \\
&\approx \alpha\left[\mathcal{R}_2\left(\tilde{U}_2^{\{k\}},\phi_2^{[1]};U_1^{\{k-1\}}\right)+\left(J_2(U_1^{\{k-1\}})e_1^{\{k-1\}},\phi_2^{[1]}\right)\right] \\
&\qquad +(1-\alpha)\left(e_2^{\{k-1\}},\psi_2^{[1]}\right) \\
&= \alpha\left[\mathcal{R}_2\left(\tilde{U}_2^{\{k\}},\phi_2^{[1]};U_1^{\{k-1\}}\right)+\left(e_1^{\{k-1\}},\psi_1^{[2]}\right)\right]+(1-\alpha)\left(e_2^{\{k-1\}},\psi_2^{[1]}\right) \\
&= \alpha\left[\mathcal{R}_2\left(\tilde{U}_2^{\{k\}},\phi_2^{[1]};U_1^{\{k-1\}}\right)+\left(e_1^{\{k-1\}},\psi_1^{[2]}\right)\right] \\
&\qquad +\alpha(1-\alpha)\left[\mathcal{R}_2\left(\tilde{U}_2^{\{k-1\}},\phi_2^{[1]};U_1^{\{k-2\}}\right)+\left(e_1^{\{k-2\}},\psi_1^{[2]}\right)\right] \\
&\qquad\quad (1-\alpha)^2\left(e_2^{\{k-2\}},\psi_2^{[1]}\right) \\
&\ \ \vdots \\
&= \alpha\sum_{i=1}^{k}(1-\alpha)^{i-1}\left[\mathcal{R}_2\left(\tilde{U}_2^{\{k-i+1\}},\phi_2^{[1]};U_1^{\{k-i\}}\right)+\left(e_1^{\{k-i\}},\psi_1^{[2]}\right)\right].
\end{aligned}
$$

In order to obtain a full *a posteriori* error estimate, we have to repeat this argument to estimate $\left(e_1^{\{k-i\}},\psi_1^{[2]}\right)$, $i=1,\ldots,k$. We refrain from doing this. Clearly repeated application of this analysis approach results in a great number of adjoint problems to be solved. It is to be hoped that introducing relaxation means that relatively few number history terms need to be included for accuracy in the estimate. As expected the decay rate of the history terms decreases as $\alpha \to 0$.

4. NUMERICAL EXAMPLES FOR FULLY COUPLED SEMILINEAR ELLIPTIC SYSTEMS

We present a number of examples to illustrate characteristics of the *a posteriori* error estimate. We also illustrate the use of the estimates as the basis to adaptively adjust discretization parameters based on relative sizes of contributions to the error estimate.

Some of the examples are used to explore the accuracy of the *a posteriori* error estimate in situations in which the operator decomposition iteration is converging well. To measure the accuracy of the error estimate, we report on the ratio of the estimate to the true error (or an approximation computed with a highly accurate reference solution). This type of *a posteriori* error estimate tends to be robustly accurate for a wide range of spatial meshes, as has been well reported in the literature. We observe the same robust accuracy in the estimates when the iteration is converging well. Rather than reporting extensively on that quality, we concentrate the experiments on situations in which the estimates do not work as well as hoped.

For adaptive error control, we employ a natural generalization of the "mark and refine" strategy based on the Principle of Equidistribution [1, 5–8]. In this approach, the error estimate is replaced by a bound obtained by replacing the signed contributions to the error estimate with their absolute values. Starting with a coarse discretization - mesh with large element size and small number of iterations - we adjust the various discretization parameters according to the relative size of the contributions and an estimate of the computational costs associated with changes in the parameters. This adaptive approach is described fully in the earlier paper [1].

We make one simplification below. In [1], we allowed the different components to have different meshes, which requires modification of the *a posteriori* error analysis to account for the effect of mesh changes on the numerical solution. Below, we use the same mesh for all components of the

DOI: 10.1002/nme

system in order to avoid introduction of the additional terms in the *a posteriori* error estimate. The results from the first paper can be combined with the results in this paper in a straightforward way.

4.1. Fully coupled linear system

The first example is a coupled system of linear equations for which we can compute an exact solution. The coupled problem is: Find $u_1(x,y), u_2(x,y)$ satisfying,

$$\begin{aligned} -\nabla\cdot(\alpha\nabla u_1) + b_1\cdot\nabla u_1 &= f_1(u_2), \quad \mathbf{x}\in\Omega, \\ -\Delta u_2 &= b_2\cdot\nabla u_1, \quad \mathbf{x}\in\Omega, \\ u_1 = u_2 &= 0, \quad \mathbf{x}\in\partial\Omega, \end{aligned} \tag{4.1}$$

where $\Omega := \{(x,y) : (x,y)\in[0,1]\times[0,1]\}$,

$$\alpha = 10 + \pi^2\left(\frac{5}{13}\cos 4\pi x + \frac{1}{2}\cos\pi y\right)$$

$$b_1 - \frac{1}{\pi}\left[\frac{8}{13}\sin 4\pi x \quad \frac{1}{2}\sin\pi y\right]^{\top}, \quad b_2 = \frac{1}{\pi}\left[\frac{1}{2}\sin 4\pi x \quad 2\sin\pi y\right]^{\top}$$

$$f_1(u_2) = 85\pi^2(2\sin 4\pi x\sin\pi y + u_2).$$

This system has exact solution

$$\begin{aligned} u_1 &= \sin 4\pi x\sin\pi y \\ u_2 &= \pi^{-2}(\sin 8\pi x\sin\pi y/65 + \sin 4\pi x\sin 2\pi y/20). \end{aligned}$$

We select the quantity of interest $u_2(x_0)$ where $x_0 = (0.15, 0.15)$; the adjoint data is then equal to δ_{x_0}, which we regularize and denote as $\delta_{x_0}^{reg}$.

As mentioned we solve for u_1 and u_2 on a common mesh, using piecewise linear finite elements. The Gauss-Seidel iteration proceeds until $\|U^{\{k\}} - U^{\{k-1\}}\|_\infty \leq 10^{-5}$. Quadratic finite elements are used to compute $\Phi_1^{[j]}$ and $\Phi_2^{[j]}$ approximations to $\phi_1^{[j]}$ and $\phi_2^{[j]}$. Starting with a uniform initial mesh, the subsequent meshes are adapted identically using the sum of just the first four terms in the error representation formula, namely, the sum of:

1. the "primary" error, $\varepsilon_1 = \mathcal{R}_2\left(U_2^{\{k\}}, \Phi_2^{[1]}; U_1^{\{k\}}\right)$,
2. the "transfer" error, $\varepsilon_2 = \mathcal{R}_1\left(U_1^{\{k\}}, \Phi_1^{[1]}; U_2^{\{k-1\}}\right)$,
3. the "inherited" error, $\varepsilon_3 = \mathcal{R}_2\left(U_2^{\{k-1\}}, \Phi_2^{[2]}; U_1^{\{k-1\}}\right)$, and
4. the "inherited transfer" error, $\varepsilon_4 = \mathcal{R}_1\left(U_1^{\{k-1\}}, \Phi_1^{[2]}; U_2^{\{k-2\}}\right)$.

The corresponding adjoint problems used to compute the weighted residuals are

$$\begin{aligned} \mathcal{A}_2^*\left(\chi, \phi_2^{[1]}\right) &= \left(\chi, \delta_{x_0}^{reg}\right), \ \forall\chi\in\mathcal{S}_2^h(\Omega), \\ \mathcal{A}_1^*\left(\chi, \phi_1^{[2]}\right) &= \left(\chi, -\mathrm{div}(b_2\phi_2^{[1]})\right), \ \forall\chi\in\mathcal{S}_2^h(\Omega), \\ \mathcal{A}_2^*\left(\chi, \phi_2^{[2]}\right) &= \left(\chi, -\mathrm{div}(b_1\phi_1^{[2]})\right), \ \forall\chi\in\mathcal{S}_2^h(\Omega), \\ \mathcal{A}_1^*\left(\chi, \phi_1^{[3]}\right) &= \left(\chi, -\mathrm{div}(b_2\phi_2^{[2]})\right), \ \forall\chi\in\mathcal{S}_2^h(\Omega). \end{aligned} \tag{4.2}$$

The initial mesh is a uniform partition with step size $h = .1$ for 200 elements. The iteration is repeated until the total error representation estimate for the quantity of interest is less than 10^{-4}. The adaptive algorithm ran for three iterations before meeting the tolerance. The final mesh has 2378 elements. The number of iterations used in Gauss-Seidel on the finest (last) mesh is 5. The iteration error estimate reports 1.43×10^{-6}.

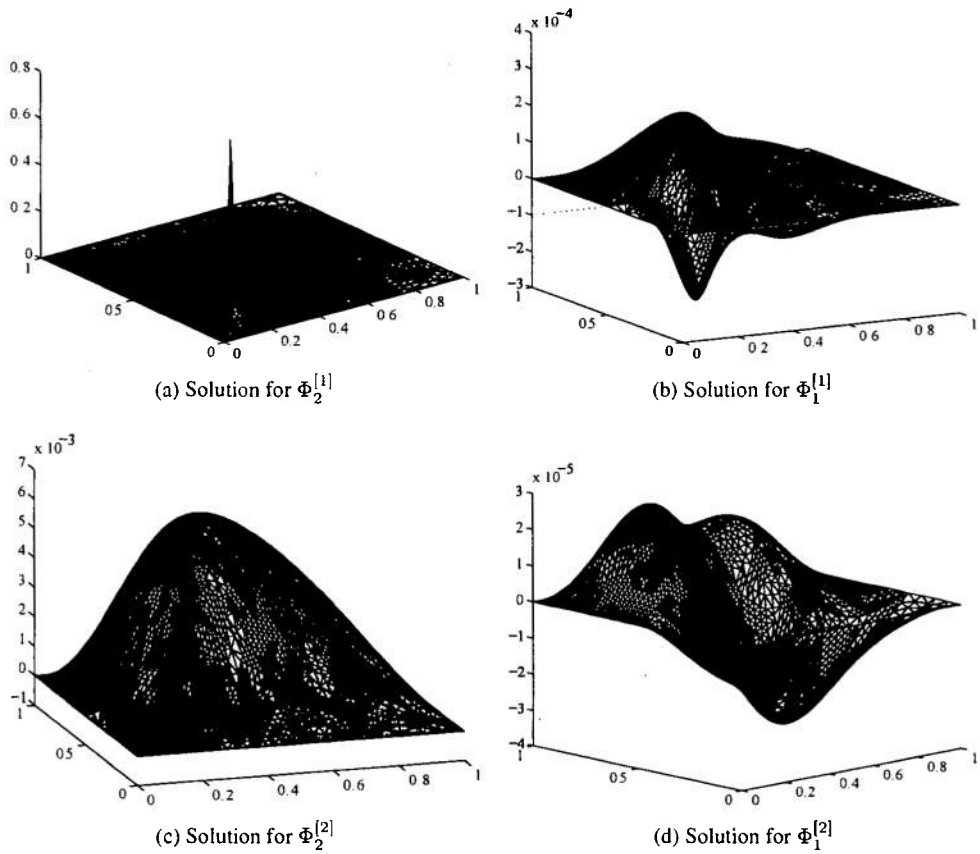
The effectivity ratio (estimate/error) is .9976.

Table 4.1 shows the contributions to the error. We observe that the "transfer" error ε_2 is about 1/5th of the size of the "primary" discretization error at iteration ε_1, and that the "inherited transfer error" ε_4 is 1/10th of the size of the primary discretization error. By contrast ε_3, the error inherited from U_2 at the iteration $(k-1)$, is 1/500th the size of ε_1.

DOI: 10.1002/nme

Primary Error (ε_1) $\mathcal{R}_2\big(U_2^{\{k\}}, \phi_2^{[1]}; U_1^{\{k\}}\big)$	Transfer Error (ε_2) $\mathcal{R}_1\big(U_1^{\{k\}}, \phi_1^{[1]}; U_2^{\{k-1\}}\big)$
0.5070×10^{-4}	0.0940×10^{-4}
Inherited Error (ε_3) $\mathcal{R}_2\big(U_2^{\{k-1\}}, \phi_2^{[2]}; U_1^{\{k-1\}}\big)$	**Inherited Transfer Error (ε_4)** $\mathcal{R}_1\big(U_1^{\{k-1\}}, \phi_1^{[2]}; U_2^{\{k-2\}}\big)$
-0.0010×10^{-4}	0.0409×10^{-4}

Table 4.1. The first four error terms, i.e., the primary error and transfer error at iteration k and the inherited error and inherited transfer error from the iteration $(k-1)$ for Ex. 4.1.



(a) Solution for $\Phi_2^{[1]}$ (b) Solution for $\Phi_1^{[1]}$

(c) Solution for $\Phi_2^{[2]}$ (d) Solution for $\Phi_1^{[2]}$

Figure 4.1. The first four adjoint solutions, i.e., the primary, transfer and the first two inherited adjoint solutions for Ex. 4.1.

4.2. Effects of iteration on the accuracy of the error estimate

We next consider a fully coupled semilinear system that is solved by the Jacobi operator decomposition iteration employing continuous piecewise linear elements with initial solutions

DOI: 10.1002/nme

$U_1^{\{0\}} = x$, $U_2^{\{0\}} = 0$. The coupled system is: Find $u_1(x), u_2(x)$ for $x \in \Omega = [0,1]$ such that

$$\begin{aligned} -u_1'' &= e^{\lambda^2 u_2}, \quad x \in \Omega, \\ -u_2'' &= \sin(\beta\pi u_1), \quad x \in \Omega, \\ u_1(0) = 0,\ u_1(1) = 1, &\qquad u_2(0) = u_2(1) = 0. \end{aligned}$$

The quantity of interest is the average value of u_2 over the whole domain. The sequence of adjoint problems are

$$\begin{aligned} -\left(\phi_1^{[j]}\right)'' &= \psi_1^{[j]}, \quad x \in \Omega, \\ -\left(\phi_2^{[j]}\right)'' &= \psi_2^{[j]}, \quad x \in \Omega, \\ \phi_1(0) = \phi_2(0) = 0, &\qquad \phi_1(1) = \phi_2(1) = 0, \end{aligned}$$

with

$$\begin{aligned} \psi_2^{[1]} &= 1 \\ \psi_2^{[j]} &= \lambda^2 e^{\lambda^2 U_2^{\{k-j\}}} \phi_1^{[j-1]}, \quad j \text{ odd} \\ \psi_1^{[j]} &= \beta\pi \cos(\beta\pi U_1^{\{k-j\}}) \phi_2^{[j-1]}, \quad j \text{ even}. \end{aligned}$$

Quadratic finite elements are used to compute $\Phi_1^{[j]}$ and $\Phi_2^{[j]}$ approximations to $\phi_1^{[j]}$ and $\phi_2^{[j]}$ and the iteration is performed until the iteration error estimator (2.11) is less than 10^{-6} or until a maximum of 30 Gauss-Seidel iterations are performed.

We examine different sets of parameters that affect the convergence of the iteration and, consequently, accuracy of the estimate.

4.2.1. Slowly decaying history contributions. The first example demonstrates problems that can arise when the history contributions decay very slowly. We fix $\beta = 10$ and compute approximations for $\lambda = 2, 4, 8, 16$. As α grows the nonlinearity becomes strong and the diagonal dominance in the iteration is weakened. We use a uniform mesh with $h = 0.05$ for the approximate solutions and we use a fine mesh with $h = .005$ to compute a "reference" solution. We run the iterations until $\|U^{\{i\}} - U^{\{i-1\}}\| \leq 10^{-6}$ up to a maximum of 30 iterations.

The iteration converges for $\lambda = 2, 4$ and 8, but for $\lambda = 16$ the iteration cannot converge (without relaxation) even using a finer space mesh. We report the error estimate effectivity (estimate/error) ratios in Table ??.

λ	ratio
2	0.999942588
4	0.997737471
8	0.978260934
16	1.647440221

Table 4.2. Effectivity ratios for Example 4.2.1.

These results the typical accuracy of the estimate. The estimate is even reasonably accurate in the last case where, as noted, the iteration does not converge. This estimate in the last case is affected by significant linearization errors since U_2 does not represent u_2 very well.

Figure 4.2 presents the partial sums of the first j history terms for the different values of λ. For $\lambda = 2, 4, 8$, the error contributions become relatively constant when more than five error terms are included, but in all cases taking simply the first error term only (the "primary error contribution") leads to a poor error estimate.

DOI: 10.1002/nme

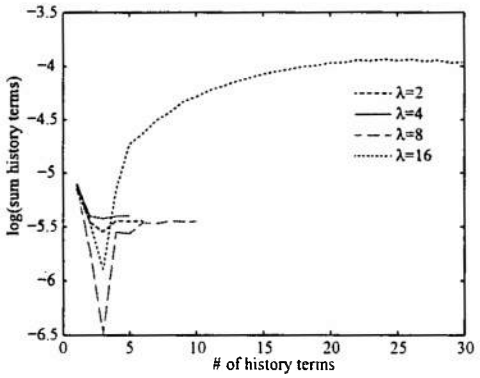


Figure 4.2. Numerical error estimate $\sum_{j=1}^{m} \mathcal{R}\left(U^{\{k-j+1\}}, \phi^{[j]}; U^{\{k-j\}}\right)$ including m history terms, for Ex. 4.2.1 and $\lambda = 2, 4, 8, 16$.

4.2.2. The effect of large numerical error on the history contributions. Let $\lambda = 10$, $\beta = 10$, $h = 0.05$. The fixed point iteration fails on a coarser mesh of $h = 0.2$, but for $h = 0.05$ the iteration converges slowly. In this computation, the error estimate is affected both by the inaccuracies due to linearizing about $U^{\{i\}}$ for i relatively small as well as the poor numerical resolution of the adjoint solution Φ (computed using quadratic finite elements on the mesh for U). Both sources of numerical error destroy the accuracy of the estimate. We see the effect in Figure 4.3, where we observe that the contributions to the error estimator *increase* as the final("older") history terms are included, after having reached an earlier plateau. This is the opposite of the expected behavior that error contributions should decay, and is in an indicator that the error estimator is not performing well. Large adjoint solutions relative to the size of the solution can be an indicator of unreliable error estimates.

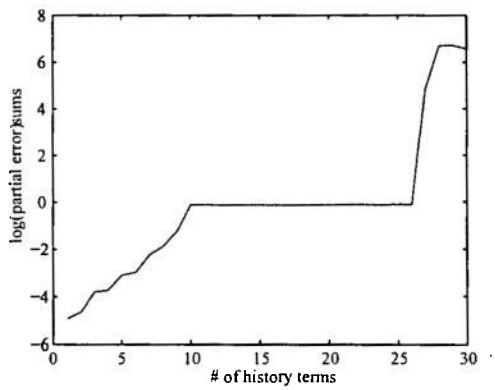


Figure 4.3. Numerical error estimate $\sum_{j=1}^{m} \mathcal{R}\left(U^{\{k-j+1\}}, \phi^{[j]}; U^{\{k-j\}}\right)$ including m history terms for Ex. 4.2.2, demonstrating that numerical error can lead to non-convergence and poor numerical error estimates.

4.2.3. The effect of a poor initial iterate on the history contributions. Let $\lambda = 10$, $\beta = 10$, $h = 0.05$. The choice of initial data for the iteration can strongly influence the performance of the error estimator. In Figure 4.4 we plot the error contributions for $U_2^{\{0\}} = 0$ versus $U_2^{\{0\}} = x(1-x)$. In this case, the norm of several of the early solution iterates is greater than 10^{14}, while the norm of the converged $U^{\{k\}}$ is $O(1)$. The difficulty is that each history adjoint problem is solved using quadratic finite elements of size $h = 0.05$, and the resulting use of $\Phi^{[j]}$ instead of $\phi^{[j]}$ in the resulting error representation formula yields an error which may be bounded by $Ch\|U^{\{k-j+1\}}\|_{W_2^1([0,1])}\|\phi^{[j]} - \Phi^{[j]}\|_{W_2^1([0,1])} = Ch^3\|U^{\{k-j+1\}}\|\|\psi^{[j]}\|$ (assuming smooth ψ, ϕ and

DOI: 10.1002/nme

u). The relative size of $U^{\{k-j+1\}}$ and $U^{\{k\}}$ means that the resulting error estimate is useless, despite the exponential decay of $\phi^{[j]}$ as j increases. Incorporating additional history terms can reduce the quality of the estimator.

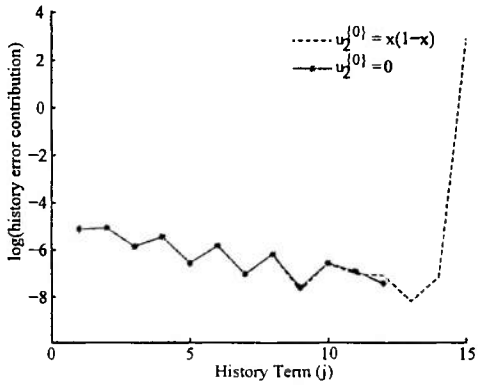


Figure 4.4. Individual "history" terms $\mathcal{R}\left(U^{\{k-j+1\}}, \phi^{[j]}; U^{\{k-j\}}\right)$ for Ex. 4.2.3, illustrating the sensitivity of history error contributions to initial data.

4.2.4. Adaptive selection of relaxation parameter and mesh resolution. In the three previous examples, problems with accuracy of the error estimate arise from slow convergence of the iteration and subsequent slow decay of the history contributions to the error. In Sec. 3.1.3, we described an extension of the *a posteriori* error estimate to a version of the Jacobi iteration that employs relaxation. The relaxation parameter directly enters into the history contributions of the error estimate.

This suggests a generalization of an adaptive strategy in which the decay of the history contributions is monitored, and the iteration is interrupted and restarted with a new relaxation parameter value if the history contributions are contributing too much relative to the other error contributions. The efficiency question is balancing the error contributions between those arising from the discretization against those arising from the iteration.

Consider once again the problem in §4.2, with quantity of interest equal to the value of u_2 at $x = 0.1$ (adjoint data $\psi_2^{[1]} = \delta_{0.1}$). When $\beta = 10$ and $\lambda = 20$ this iteration cannot converge without relaxation, regardless of mesh density.

We present the adaptive algorithm in Alg. 4. We begin the adaptive procedure using a coarse common mesh for both U_1 and U_2 with $h = .2$. We perform a Jacobi iteration with no relaxation until the convergence criterion ($\|U^i - U^{i-1}\|_\infty \leq 10^{-7}$) is satisfied or a maximum of 8 iterations is reached. At this point, the (computable) iteration error estimate (2.11) is calculated. We then compute the error representation formula (3.4) using a variable number of history terms. In general, the minimum number of history terms is set to be $\min(4, k)$, but the number of history terms computed is halted if the jth history term is too small ($< C_1 h \mathcal{E}_{tol}$) or too large ($> C_2 h^{-1} \mathcal{E}_{tol}$) where $C_1 = 0.1$ and $C_2 = 10$. We adapt the common mesh for U_1 and U_2 using a standard "mark and refine" strategy applied to the *a posteriori* estimate using ||local error|| <TOL/(# of elements) as a marking criterion. If the iteration does not "converge" or if the iteration error estimate is greater than the numerical error estimate, we increase the relaxation parameter α and repeat the process. Various choices for updating α cam be employed at this step. The process continues until both the iteration and numerical error estimate are less than 10^{-5}.

We show the final adaptive solution for U_2 in Figure 4.5. Generally, a quantity of interest of the value at a point leads to a mesh that is highly refined in a local neighborhood of the point. But the final refined mesh is nearly uniform as we see. This is a consequence of the errors inherited from previous iterations coupled to the fact that the solutions have significantly different scales, i.e. U_1 is $\mathcal{O}(1)$ while U_2 is $\mathcal{O}(10^-3)$). Thus, the localized contributions that should result in the refinement of the mesh for U_2 near $x = .1$ are masked by the much larger - and nearly uniformly distributed - contributions from U_1 in previous iterations.

DOI: 10.1002/nme

Algorithm 4 Adaptive algorithm including relaxation adjustment

while total error < TOL **do**
  while iterations $\leq k$ OR it. error > TOL **do**
    Compute $U^{\{i+1\}}$
  end while
  Compute $\Phi^{[1]}, \ldots, \Phi^{[m]}$, $1 \leq m < i$
  while $m < i$ or not STOP **do**
    if mth error term $> C_1 h \sum_1^{m-1}$ (error terms) **then**
      if mth error term $< C_2 h^{-1} \sum_1^{m-1}$ (error terms) **then**
        $m = m + 1$
        Compute $\Phi^{[m+1]}$
      else
        STOP=true
      end if
    else
      STOP=true
    end if
  end while
  Refine mesh using Non-Iteration error estimate
  if Iteration error > Non-Iteration error **then**
    update relaxation parameter α.
  end if
end while

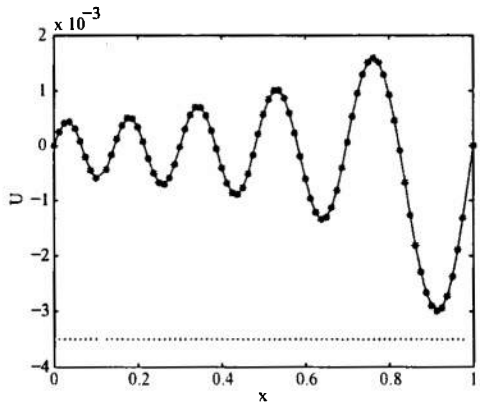


Figure 4.5. Final adapted solution for U_2 for Ex. 4.2.4 showing the spatial grid.

5. CONCLUSIONS

We have presented an *a posteriori* error analysis for operator decomposition iterative solution of systems of coupled semilinear elliptic systems that use a block iterative solution technique. The analysis provides the means to compute accurate error estimates that account for discretization errors in the solution of each component at a given iteration, the errors passed between components at a given iteration, numerical errors inherited from previous iterations and errors arising due to the iterative solution procedure. This paper specifically addresses the propagation of error between iterates in the operator decomposition iteration solution and the effects of finite iteration on the error estimate. We extend the adaptive discretization strategy in [1] to systematically reduce the error.

DOI: 10.1002/nme

REFERENCES

1. Carey V, Estep D, Tavener S. *A posteriori* analysis and adaptive error control for operator decomposition solution of elliptic systems I: Triangular systems. *SIAM Journal of Numerical Analysis* 2009; **47**:740–761.
2. Estep D. Error estimation for multiscale operator decomposition for multiphysics problems. *Bridging the Scales in Science and Engineering*, Fish J (ed.). chap. 11, Oxford University Press, 2010.
3. Ainsworth M, Oden J. *A posteriori* error estimation in finite element analysis. *Comput. Method. Appl. M.* 1997; **142**(1–2):1–88.
4. Heuveline V, Rannacher R. Duality-based adaptivity in the hp-finite element method. *J. Numer. Math* 2003; **1**:95–113.
5. Giles M, Süli E. Adjoint methods for PDEs: *A posteriori* error analysis and postprocessing by duality. *Acta Numerica, 2002*. Acta Numer., Cambridge Univ. Press: Cambridge, 2002; 105–158.
6. Becker R, Rannacher R. An optimal control approach to a posteriori error estimation in finite element methods. *Acta Numerica, 2001*. Acta Numer., Cambridge Univ. Press: Cambridge, 2001; 1–102.
7. Eriksson K, Estep D, Hansbo P, Johnson C. Introduction to adaptive methods for differential equations. *Acta Numerica, 1995*. Acta Numer., Cambridge Univ. Press: Cambridge, 1995; 105–158.
8. Estep D, Larson MG, Williams RD. Estimating the error of numerical solutions of systems of reaction-diffusion equations. *Mem. Amer. Math. Soc.* 2000; **146**(696):viii+109.
9. Larson MG, Bengzon F. Adaptive finite element approximation of multiphysics problems. *Communications in Numerical Methods in Engineering* 2008; **24**:505–521.
10. Marchuk G, Agoshkov V, Shuyaev V. *Adjoint Equations and Perturbation Algorithms*. CRC Press: New York, 1996.

DOI: 10.1002/nme

IMA Journal of Numerical Analysis (2013) Page 1 of 24
doi:10.1093/imanum/drnxxx

A-posteriori error estimates for mixed finite element and finite volume methods for problems coupled through a boundary with non-matching grids

T. Arbogast †, *D. Estep*‡, *B. Sheehan*§ *and S. Tavener*¶

[Received on 12 March 2013]

Using an adjoint based a-posteriori error estimate, we explore the accuracy of two different discretization schemes applied to elliptic problems in which there are different meshes on two neighboring subdomains that share an interface. The first discretization is a mixed finite element mortar method which relies on interface variables to couple the subdomains, while the second discretization is a finite volume method which relies on geometrically motivated projections to couple the subdomains. To facilitate comparison of the accuracy of the two methods using the a-posteriori error estimate, the finite volume method is cast as a mixed finite element method using appropriate quadrature. The a-posteriori error estimate is derived and used to analyze both the size and source of the discretization error of both methods. We identify, through numerical examples, cases in which the geometric projections are the dominant source of error by one to two orders of magnitude. While this effect may be expected in examples where the solution is changing rapidly near the interface, it is also demonstrated for an example in which the solution is smooth, and nearly one dimensional across the interface.

Keywords: mortar methods, a-posteriori error estimate, coupled elliptic problems, heterogeneous domain decomposition, geometric coupling

1. Introduction

An important class of multiphysics problems has a structure in which one physical process dominates in one subdomain of the problem domain, while a second physical process dominates in a neighboring subdomain. The solutions are coupled by continuity of state and continuity of normal flux through a shared boundary between the subdomains. Examples include general problems of the heterogeneous domain decomposition type (Quarteroni *et al.*, 1992; Gaiffe *et al.*, 2002; Bernardi *et al.*, 1994), core-edge plasma simulations of a tokamak fusion experiment (Cary *et al.*, 2008, 2010), and conjugate heat transfer between a fluid and solid object (Estep *et al.*, 2008, 2009b, 2010).

†Department of Mathematics, University of Texas at Austin (arbogast@math.utexas.edu). T. Arbogast's work was supported as part of the Center for Frontiers of Subsurface Energy Security, an Energy Frontier Research Center funded by the U.S. Department of Energy, Office of Science, Office of Basic Energy Sciences under Award Number DE-SC0001114.

‡Department of Statistics, Colorado State University, Fort Collins, CO 80523 (estep@stat.colostate.edu). D. Estep's work is supported in part by the Defense Threat Reduction Agency (HDTRA1-09-1-0036), Department of Energy (DE-FG02-04ER25620, DE-FG02-05ER25699, DE-FC02-07ER54909, DE-SC0001724, DE-SC0005304, INL00120133), Idaho National Laboratory (00069249, 00115474), Lawrence Livermore National Laboratory (B584647, B590495), National Science Foundation (DMS-0107832, DMS-0715135, DGE-0221595003, MSPA-CSE-0434354, ECCS-0700559, DMS-1065046, DMS-1016268, DMS-FRG-1065046), National Institutes of Health (#R01GM096192).

§Department of Mathematics, Colorado State University (brendansheehan6@gmail.com).

¶Department of Mathematics, Colorado State University.

In such situations, it is common to encounter significant differences in scales of behavior in the two subdomains. This in turn suggests the use of different discretization grids. However, this introduces the problem of interpreting the meaning of coupling state and flux values through the common boundary in the discretization, since exact pointwise matching is no longer possible.

Confounding this issue are the practical difficulties of solving the large linear and nonlinear discrete systems associated with computing numerical solutions and the common situation in which two different codes are used to solve the two subdomain problems. These difficulties are generally tackled by employing some form of iterative approach that involves sequential solution of the subdomain problems. The particular properties of the discretizations used for each component problem, the choice of iterative solution method, and high performance computational considerations all have a large impact on the way in which state and flux values are passed across the common interface.

In this paper, we investigate the accuracy of two approaches to computing the coupling values in the situation in which the discretization grids in the two subdomains do not match at the interface. The analysis is carried out for the closely related mixed finite element and cell-centered finite volume methods. The two approaches are (1) the mortar element approach (Brezzi & Fortin, 1991; Roberts & Thomas, 1991; Arbogast *et al.*, 2000; Ben Belgacem, 2000; Arbogast *et al.*, 2007; Ganis & Yotov, 2009), which uses a rigorous variational formulation to define a weak sense of coupling, and (2) a "geometric" approach that employs various ad hoc extrapolation and averaging methods. The use of mortar elements is proven to be optimally convergent on nonmatching grids, provided the finite element space used for the interface variables consists of piecewise polynomials of one degree higher than the trace along the interface of the finite element space used to approximate the flux within the subdomains (Arbogast *et al.*, 2000). Nonetheless, while mortar elements are well known in some application domains, e.g., flow in porous media, they are not widely employed for multiphysics problems. Rather, various "geometric" techniques are used in most practical settings, especially in situations in which one or more of the components are solved with legacy "black box" codes. This second approach is often rationalized using a combination of ad hoc formal stability and/or accuracy arguments combined with high performance computing expediences. Moreover, in the situation in which legacy codes are used to solve either component, there is little choice because of the very considerable investment that would be required to replace these codes.

We are not arguing for or against either mortar elements or "geometric" approaches. Rather, we address two issues: (1) What effect do these coupling approaches have on accuracy of specified quantities of interest? and (2) In each case, quantify the relative contributions of various aspects of discretization to the error in the computed information. The tool we use to address these issues is an adjoint-based a-posteriori error estimate (Estep *et al.*, 2000; Becker & Rannacher, 2001; Giles & Suli, 2002; Wheeler & Yotov, 2005; Estep *et al.*, 2009a; Hansbro & Larson, 2011). This goal-oriented estimate accurately quantifies various contributions to the overall error. In particular, the estimate distinguishes contributions specifically arising from the mis-matched grids and the way in which the coupled information is approximated. We identify, through numerical examples, cases in which the geometric projections are the dominant source of error by one to two orders of magnitude.

The remainder of this paper is organized as follows. Section two introduces the continuous problem and the details of the two discrete methods. Section three derives the a-posteriori error estimate. Section four contains the numerical experiments. Section five discusses computational logistics related to iterative solvers, and a brief conclusion is given in section six.

2. Definition of the problem and discretization methods

We define the coupled problem with a common interface, then describe the finite element and finite volume discretizations. We employ the well known equivalence between finite volume methods and the mixed finite element method (Russell & Wheeler, 1983; Weiser & Wheeler, 1988) to recast everything in the finite element framework. This greatly eases the derivation of a-posteriori error estimates and provides a systematic framework for describing geometric approaches to computing coupling values.

2.1 *The continuous problem*

The continuous problem (2.1)–(2.3) consists of a system of second order elliptic partial differential equations (PDE) in two spatial dimensions. The system is posed on a rectangular domain Ω consisting of two nonoverlapping rectangular subdomains, Ω_L on the left-hand side and Ω_R on the right-hand side, that share a common interface Γ_I, and whose union forms the entire domain, as shown in Fig. 1. The unit normal vector $\boldsymbol{n}$ is defined to point from left to right on Γ_I, and is an outward pointing normal on $\Gamma_L = \partial\Omega_L \setminus \Gamma_I$ and $\Gamma_R = \partial\Omega_R \setminus \Gamma_I$. For simplicity of presentation, we assume Dirichlet boundary conditions on $\partial\Omega$, the external boundaries of the domain. The results extend to problems with Neumann conditions on part of the boundary in a straightforward way.

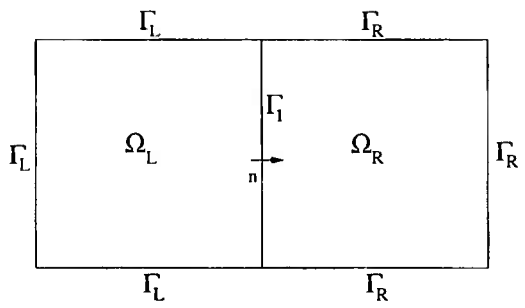


FIG. 1. Subdomains, boundaries, and definition of normal $\boldsymbol{n}$ on the interface.

For a source function f, split as $f_L \in L^2(\Omega_L)$ and $f_R \in L^2(\Omega_R)$, and boundary data g, similarly split as $g_L \in H^{3/2}(\Gamma_L)$ and $g_R \in H^{3/2}(\Gamma_R)$, the coupled system is

$$\begin{cases} a^{-1}\boldsymbol{u}_L + \nabla p_L = 0, & (x,y) \in \Omega_L, \\ \nabla \cdot \boldsymbol{u}_L = f_L, & (x,y) \in \Omega_L, \\ p_L = g_L, & (x,y) \in \Gamma_L, \end{cases} \tag{2.1}$$

$$\begin{cases} a^{-1}\boldsymbol{u}_R + \nabla p_R = 0, & (x,y) \in \Omega_R, \\ \nabla \cdot \boldsymbol{u}_R = f_R, & (x,y) \in \Omega_R, \\ p_R = g_R, & (x,y) \in \Gamma_R, \end{cases} \tag{2.2}$$

$$\begin{cases} \xi \equiv p_L = p_R, & (x,y) \in \Gamma_I, \\ \boldsymbol{n} \cdot (\boldsymbol{u}_L - \boldsymbol{u}_R) = 0, & (x,y) \in \Gamma_I, \end{cases} \tag{2.3}$$

where we assume that the diffusion matrix, a, is a function of space times the identity,

$$a = \begin{bmatrix} D(x,y) & 0 \\ 0 & D(x,y) \end{bmatrix}, \tag{2.4}$$

with $D \in W^{1,\infty}(\Omega)$ and $\min_{(x,y)\in\overline{\Omega}} D(x,y) \geqslant D_0 > 0$, so a is invertible and uniformly coercive. Note that we have defined ξ as the common interface pressure in (2.3).

2.2 *Mixed finite element mortar discretization*

The mortar finite element discretization was developed precisely for the situation presented by discretization of (2.1)–(2.3) using two different grids in the two different subdomains. We assume that each subdomain is discretized by a (logically) rectangular finite element grid. Lagrange multipliers are introduced on the interface boundary to provide a weak formulation of the pressure coupling conditions. Since the grids are different on the two sides of the interface, the Lagrange multiplier space cannot be the normal trace of the velocity space. So, we introduce a mortar finite element space on the interface (Arbogast *et al.*, 2000; Bernardi *et al.*, 2005; Arbogast *et al.*, 2007). As shown in Arbogast *et al.* (2000), the method is optimally convergent and has several other desirable convergence properties if the boundary space has one order higher approximability than the normal trace of the velocity space. The same order of convergence is obtained for both continuous or discontinuous piecewise polynomials in the mortar space. In our discretization, we choose the interface grid that has one cell for every two cells in the finer of the two subdomain grids. Fig. 2 shows the arrangement for a 5×5 grid next to 8×8 grid. (Note that our convention is that the finer grid is always used in the righthand subdomain.)

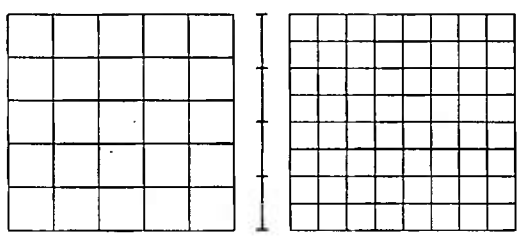


FIG. 2. Example grid shown separated into the part on Ω_L, Γ_I, and Ω_R from left to right.

We use standard L^2 inner product notation, i.e., for functions F and G defined on Ω, split as above,

$$(F_i, G_i) = \int_{\Omega_i} F_i(x,y)\, G_i(x,y)\, dx\, dy, \quad i = L, R,$$

and for functions defined on the boundaries, we similarly denote

$$\langle F, G\rangle_{\Gamma_i} = \int_{\Gamma_i} F\, G\, ds, \quad i = L, I, R.$$

The mixed finite element (mortar) method starts with the following continuous weak formulation. Find

$p_i \in W_i = L^2(\Omega_i)$, $\boldsymbol{u}_i \in \boldsymbol{V}_i = H(\mathrm{div};\Omega_i)$, $\xi \in \Lambda = H^{1/2}(\Gamma_I)$, $i = L,R$, satisfying

$$\begin{aligned}
&(a^{-1}\boldsymbol{u}_L, \boldsymbol{v}_L) - (p_L, \nabla\cdot\boldsymbol{v}_L) + \langle \xi, \boldsymbol{n}\cdot\boldsymbol{v}_L\rangle_{\Gamma_I} = -\langle g_L, \boldsymbol{n}\cdot\boldsymbol{v}_L\rangle_{\Gamma_L},\\
&(\nabla\cdot\boldsymbol{u}_L, w_L) = (f_L, w_L),\\
&(a^{-1}\boldsymbol{u}_R, \boldsymbol{v}_R) - (p_R, \nabla\cdot\boldsymbol{v}_R) - \langle \xi, \boldsymbol{n}\cdot\boldsymbol{v}_R\rangle_{\Gamma_I} = -\langle g_R, \boldsymbol{n}\cdot\boldsymbol{v}_R\rangle_{\Gamma_R},\\
&(\nabla\cdot\boldsymbol{u}_R, w_R) = (f_R, w_R),\\
&\langle \boldsymbol{n}\cdot(\boldsymbol{u}_L - \boldsymbol{u}_R), v\rangle_{\Gamma_I} = 0,
\end{aligned} \tag{2.5}$$

for all $(w_i, \boldsymbol{v}_i, v) \in (W_i, \boldsymbol{V}_i, \Lambda)$, $\quad i = L,R$.

To discretize, we use the lowest order Raviart-Thomas finite element space ($RT0$), in which the discrete scalar unknown p^h is approximated as a constant over each cell, and the components of the discrete vector $\boldsymbol{u}^h$ are approximated by functions that are piecewise linear in one spatial dimension and constant in the other (Bernardi *et al.*, 2005; Estep *et al.*, 2009a). The discrete interface unknown, ξ^h, is represented by piecewise discontinuous linears on the interface grid cells (Arbogast *et al.*, 2000, 2007). The test functions in the discretization of the weak formulation of (2.5) corresponding to w, $\boldsymbol{v}$, and v are restricted to these same spaces. To be precise, for a finite element partition Δ of $[a,b]$, and for $r = 0,1,2,\ldots$, $q = -1,0,1,\ldots$, we define the piecewise polynomial space

$$\begin{aligned}
\mathscr{M}_q^r(\Delta) = \{ &v \in C^q([a,b]) :\\
&v \text{ is a polynomial of degree } \leqslant r \text{ on each subinterval of } \Delta\}.
\end{aligned}$$

When $q = -1$ the functions are discontinuous. The space of continuous piecewise bilinear functions is the tensor product $\mathscr{M}_0^1(\Delta_x)\otimes\mathscr{M}_0^1(\Delta_y)$. The RT0 discrete spaces are

$$\begin{aligned}
W_i^h &= \mathscr{M}_{-1}^0(\Delta_{x,i})\otimes\mathscr{M}_{-1}^0(\Delta_{y,i}), \quad i = L,R,\\
\boldsymbol{V}_i^h &= [\mathscr{M}_0^1(\Delta_{x,i})\otimes\mathscr{M}_{-1}^0(\Delta_{y,i})]\times[\mathscr{M}_{-1}^0(\Delta_{x,i})\otimes\mathscr{M}_0^1(\Delta_{y,i})], \quad i = L,R,\\
\Lambda^h &= \mathscr{M}_{-1}^1(\Delta_\Gamma).
\end{aligned}$$

The mixed finite element (mortar) method reads: Compute $p_i^h \in W_i^h$, $\boldsymbol{u}_i^h \in \boldsymbol{V}_i^h$, $\xi^h \in \Lambda^h$, $i = L,R$, satisfying

$$\begin{aligned}
&(a^{-1}\boldsymbol{u}_L^h, \boldsymbol{v}_L) - (p_L^h, \nabla\cdot\boldsymbol{v}_L) + \langle \xi^h, \boldsymbol{n}\cdot\boldsymbol{v}_L\rangle_{\Gamma_I} = -\langle g_L, \boldsymbol{n}\cdot\boldsymbol{v}_L\rangle_{\Gamma_L},\\
&(\nabla\cdot\boldsymbol{u}_L^h, w_L) = (f_L, w_L),\\
&(a^{-1}\boldsymbol{u}_R^h, \boldsymbol{v}_R) - (p_R^h, \nabla\cdot\boldsymbol{v}_R) - \langle \xi^h, \boldsymbol{n}\cdot\boldsymbol{v}_R\rangle_{\Gamma_I} = -\langle g_R, \boldsymbol{n}\cdot\boldsymbol{v}_R\rangle_{\Gamma_R},\\
&(\nabla\cdot\boldsymbol{u}_R^h, w_R) = (f_R, w_R),\\
&\langle \boldsymbol{n}\cdot(\boldsymbol{u}_L^h - \boldsymbol{u}_R^h), v\rangle_{\Gamma_I} = 0,
\end{aligned} \tag{2.6}$$

for all $(w_i, \boldsymbol{v}_i, v) \in (W_i^h, \boldsymbol{V}_i^h, \Lambda^h)$, $i = L,R$. This yields a discrete system of the form

$$\begin{bmatrix}
M_L & -B_L & 0 & 0 & C_L\\
B_L^T & 0 & 0 & 0 & 0\\
0 & 0 & M_R & -B_R & C_R\\
0 & 0 & B_R^T & 0 & 0\\
C_L^T & 0 & C_R^T & 0 & 0
\end{bmatrix}
\begin{bmatrix} u_L^h\\ p_L^h\\ u_R^h\\ p_R^h\\ \xi^h \end{bmatrix}
=
\begin{bmatrix} -D_L\\ F_L\\ -D_R\\ F_R\\ 0 \end{bmatrix}, \tag{2.7}$$

where we abuse notation to let u_i^h, p_i^h, and ξ^h denote the vector of nodal values for the finite element functions.

2.3 *Geometrically coupled finite volume discretization*

The standard formulation of the finite volume method eschews a variational formulation of the problem, so there is no natural description of a weak imposition of the coupling conditions in that formulation. Moreover, the standard finite volume method provides approximation values of p only at cell centers while approximate values for u along cell boundaries are obtained by differencing the p values. These characteristics motivate the use of "geometric" coupling techniques that employ a combination of extrapolation and averaging to provide coupling values of both unknowns along the interface. The motivation for this approach is reinforced in the context of iterative solution of the coupled problems, where well posed problems are created on each subdomain using interface boundary conditions obtained from the other subdomain. In this approach, it is necessary to couple the coarser side using state values extrapolated from the finer side solution, while the finer side must be coupled to flux values, which are themselves differences of state values, extrapolated from the coarser solution. Reversing this arrangement can lead to a singular system.

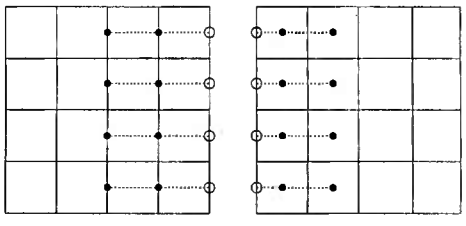


FIG. 3. Extrapolation to the interface. Left: Neumann values on the interface are computed by linear extrapolation of the last two available flux values, which are differences of state values. Right: Dirichlet values on the interface are computed by linear extrapolation of the last two available state values.

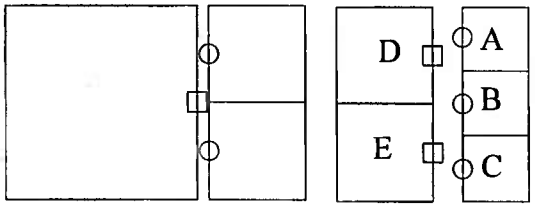


FIG. 4. Averaging or broadcasting of extrapolated values. Left: In the case of constant extrapolation, the last available state or flux value is simply used as the interface value. Right: Weighted averaging of state and flux values when cell widths do not share an integer ratio.

To obtain values on the interface, we employ either linear or constant extrapolation. We illustrate linear extrapolation in Fig. 3. We compute the extrapolated values by computing a linear or constant interpolant, which is then evaluated at the interface boundary. We denote the extrapolated values using the operators $P_{R\to L}(p_R^h)$ and $P_{L\to R}(p_L^h)$. When the cells on either side of the interface do not match, then weighted averaging and "broadcasting" schemes are used to generate values. In Fig. 4, we illustrate the averaging and broadcasting schemes when two cells on the right match one cell on the left. The state values at the two circle locations are averaged and used at the square location. The flux value at the square location is "broadcast" to both of the circle locations. When the cell widths on the coarse and fine side of the interface do not share an integer ratio, then a suitable averaging of values is used. For example, in the 2 cells next to 3 cells arrangement pictured in Fig. 4, the state value at location D is set equal to $\frac{2}{3}$ the state value at location A plus $\frac{1}{3}$ the state value at location B. The flux value at location A is set equal to the flux value at location D, while the flux value used at location B is set equal to half the flux value at D plus half the flux value at E.

We formulate the finite volume method as an RT0 mixed finite element method employing a special quadrature formula, following Russell & Wheeler (1983); Weiser & Wheeler (1988). This provides a foundation for deriving an a-posteriori error analysis for the finite volume scheme, see Estep *et al.*

(2009a). The version of (2.6) equivalent to a finite volume method reads: Compute $p_i^h \in W_i^h$, $\boldsymbol{u}_i^h \in \boldsymbol{V}_i^h$, $\xi^h \in \Lambda^h$, $i = L, R$, satisfying

$$\begin{aligned}
&(a^{-1}\boldsymbol{u}_L^h, \boldsymbol{v}_L)_{M,T} - (p_L^h, \nabla\cdot\boldsymbol{v}_L) + \langle P_{R\to L}(p_R^h), \boldsymbol{n}\cdot\boldsymbol{v}_L\rangle_{\Gamma_l} = -\langle g_L, \boldsymbol{n}\cdot\boldsymbol{v}_L\rangle_{\Gamma_L,M},\\
&(\nabla\cdot u_{h_L}, w_L) = (f_L, w_L),\\
&(a^{-1}\boldsymbol{u}_R^h, \boldsymbol{v}_R)_{M,T} - (p_R^h, \nabla\cdot\boldsymbol{v}_R) - \langle \xi^h, \boldsymbol{n}\cdot\boldsymbol{v}_R\rangle_{\Gamma_l} = -\langle g_R, \boldsymbol{n}\cdot\boldsymbol{v}_R\rangle_{\Gamma_R,M},\\
&(\nabla\cdot\boldsymbol{u}_R^h, w_R) = \langle f_R, w_R\rangle,\\
&\langle (P_{L\to R}(p_L^h) - \boldsymbol{n}\cdot\boldsymbol{u}_R^h), \nu\rangle_{\Gamma_l} = 0,
\end{aligned} \tag{2.8}$$

for all $(w_i, \boldsymbol{v}_i, \nu) \in (W_i^h, \boldsymbol{V}_i^h, \Lambda^h)$, $i = L, R$. Here we employ the approximate inner product

$$(\boldsymbol{u}^h, \boldsymbol{v})_{M,T} = (u_x^h, v_x)_{T_x,M_y} + (u_y^h, v_y)_{M_x,T_y},$$

where $M_{(\cdot)}$ and $T_{(\cdot)}$ denote the midpoint and trapezoidal quadrature rules in the x and y directions as indicated, while $\langle\cdot,\cdot\rangle_{\Gamma_i,M}$ denotes the midpoint rule for $i = L, R$.

This yields a discrete system of the form

$$\begin{bmatrix} M_L & -B_L & 0 & Q_D & 0\\ B_L^T & 0 & 0 & 0 & 0\\ 0 & 0 & M_R & -B_R & C_R\\ 0 & 0 & B_R^T & 0 & 0\\ 0 & Q_N & C_R^T & 0 & 0 \end{bmatrix} \begin{bmatrix} u_L^h\\ p_L^h\\ u_R^h\\ p_R^h\\ \xi^h \end{bmatrix} = \begin{bmatrix} -D_L\\ F_L\\ -D_R\\ F_R\\ 0 \end{bmatrix}, \tag{2.9}$$

which should be compared to (2.7).

It is possible to eliminate the unknowns u_i^h, $i = L, R$, and ξ^h, to reduce (2.9) to a system for p_i^h of the form

$$\begin{bmatrix} A_L & C_D\\ C_N & A_R \end{bmatrix} \begin{bmatrix} p_L^h\\ p_R^h \end{bmatrix} = \begin{bmatrix} F_L\\ F_R \end{bmatrix}. \tag{2.10}$$

The averaging and broadcasting are incorporated into the "coupling Dirichlet" and "coupling Neumann" matrices C_D and C_N. This is the same system that is constructed by using a finite volume approach directly.

We have verified through numerical experiments that the p component of the solution of (2.9) is identical to the solution of (2.10). Furthermore, the u component of the solution of (2.9) is identical to the u values obtained by differencing the solution of (2.10) to approximate ∇p at the cell boundaries and evaluating the diffusivity at the cell boundaries. The ξ component of the solution of (2.9) has no counterpart in the solution of (2.10).

3. A-posteriori error analysis

Our goal is to derive an a-posteriori error estimate for the quantity of interest

$$(e_{\boldsymbol{u}_L}, \boldsymbol{\psi}_{\boldsymbol{u}_L}) + (e_{p_L}, \psi_{p_L}) + (e_{\boldsymbol{u}_R}, \boldsymbol{\psi}_{\boldsymbol{u}_R}) + (e_{p_R}, \psi_{p_R}) + \langle e_\xi, \psi_\xi\rangle_{\Gamma_l}, \tag{3.1}$$

where $\boldsymbol{\psi}_{\boldsymbol{u}_L}, \psi_{p_L}, \boldsymbol{\psi}_{\boldsymbol{u}_R}, \psi_{p_R}$, and ψ_ξ are given L^2 functions and $e_{(\cdot)}$ denotes the errors in the corresponding variables. We define the generalized Green's function corresponding to these functionals using the

adjoint problem

$$\begin{cases} a^{-1}\boldsymbol{\phi}_L - \nabla\zeta_L = \psi_{\boldsymbol{u}_L} & \text{on } \Omega_L, \\ -\nabla\cdot\boldsymbol{\phi}_L = \psi_{p_L} & \text{on } \Omega_L, \\ \zeta_L = 0 & \text{on } \Gamma_L, \end{cases} \tag{3.2}$$

$$\begin{cases} a^{-1}\boldsymbol{\phi}_R - \nabla\zeta_R = \psi_{\boldsymbol{u}_R} & \text{on } \Omega_R, \\ -\nabla\cdot\boldsymbol{\phi}_R = \psi_{p_R} & \text{on } \Omega_R, \\ \zeta_R = 0 & \text{on } \Gamma_R, \end{cases} \tag{3.3}$$

$$\begin{cases} \beta \equiv \zeta_L = \zeta_R & \text{on } \Gamma_I, \\ \boldsymbol{n}\cdot(\boldsymbol{\phi}_L - \boldsymbol{\phi}_R) = \psi_\xi & \text{on } \Gamma_I. \end{cases} \tag{3.4}$$

The a-posteriori error estimates explicitly depend on $\boldsymbol{\phi}_L, \zeta_L, \boldsymbol{\phi}_R$, and ζ_R.

3.1 *Estimate for mortar mixed finite element method*

We first derive an estimate for the mortar finite element method assuming all integrals in the weak formulation are computed exactly. We begin by substituting (3.2)–(3.4) for the various ψ's in (3.1) and applying the divergence theorem,

$$\begin{aligned} &(e_{\boldsymbol{u}_L}, \boldsymbol{\psi}_{\boldsymbol{u}_L}) + (e_{p_L}, \psi_{p_L}) + (e_{\boldsymbol{u}_R}, \boldsymbol{\psi}_{\boldsymbol{u}_R}) + (e_{p_R}, \psi_{p_R}) + \langle e_\xi, \psi_\xi\rangle_{\Gamma_I} \\ &\quad = (e_{\boldsymbol{u}_L}, a^{-1}\boldsymbol{\phi}_L) + (\nabla\cdot e_{\boldsymbol{u}_L}, \zeta_L) - \langle \boldsymbol{n}\cdot e_{\boldsymbol{u}_L}, \beta\rangle_{\Gamma_I} - (e_{p_L}, \nabla\cdot\boldsymbol{\phi}_L) \\ &\qquad (e_{\boldsymbol{u}_R}, a^{-1}\boldsymbol{\phi}_R) + (\nabla\cdot e_{\boldsymbol{u}_R}, \zeta_R) + \langle \boldsymbol{n}\cdot e_{\boldsymbol{u}_R}, \beta\rangle_{\Gamma_I} - (e_{p_R}, \nabla\cdot\boldsymbol{\phi}_R) \\ &\qquad\quad + \langle e_\xi, \boldsymbol{n}\cdot(\boldsymbol{\phi}_L - \boldsymbol{\phi}_R)\rangle_{\Gamma_I}. \end{aligned} \tag{3.5}$$

Expanding on the right and subtracting

$$\begin{aligned} &(a^{-1}\boldsymbol{u}_L, \boldsymbol{\phi}_L) - (p_L, \nabla\cdot\boldsymbol{\phi}_L) + \langle \xi, \boldsymbol{n}\cdot\boldsymbol{\phi}_L\rangle_{\Gamma_I} + \langle g_L, \boldsymbol{n}\cdot\boldsymbol{\phi}_L\rangle_{\Gamma_L} \\ &\quad + (\nabla\cdot\boldsymbol{u}_L, \zeta_L) - (f_L, \zeta_L) \\ &\qquad + (a^{-1}\boldsymbol{u}_R, \boldsymbol{\phi}_R) - (p_R, \nabla\cdot\boldsymbol{\phi}_R) - \langle \xi, \boldsymbol{n}\cdot\boldsymbol{\phi}_R\rangle_{\Gamma_I} + \langle g_R, \boldsymbol{n}\cdot\boldsymbol{\phi}_R\rangle_{\Gamma_R} \\ &\qquad\quad + (\nabla\cdot\boldsymbol{u}_R, \zeta_R) - (f_R, \zeta_R) \\ &\qquad\qquad - \langle \boldsymbol{n}\cdot(\boldsymbol{u}_L - \boldsymbol{u}_R), \beta\rangle_{\Gamma_I} = 0, \end{aligned}$$

obtained by substituting the adjoint solution as test functions into the forward weak form (2.5), gives

$$\begin{aligned} &(e_{\boldsymbol{u}_L}, \boldsymbol{\psi}_{\boldsymbol{u}_L}) + (e_{p_L}, \psi_{p_L}) + (e_{\boldsymbol{u}_R}, \boldsymbol{\psi}_{\boldsymbol{u}_R}) + (e_{p_R}, \psi_{p_R}) + \langle e_\xi, \psi_\xi\rangle_{\Gamma_I} \\ &\quad = -(a^{-1}\boldsymbol{u}_L^h, \boldsymbol{\phi}_L) + (p_L^h, \nabla\cdot\boldsymbol{\phi}_L) - \langle g_L, \boldsymbol{n}\cdot\boldsymbol{\phi}_L\rangle_{\Gamma_L} - \langle \xi^h, \boldsymbol{n}\cdot\boldsymbol{\phi}_L\rangle_{\Gamma_I} \\ &\qquad + (f_L, \zeta_L) - (\nabla\cdot\boldsymbol{u}_L^h, \zeta_L) \\ &\qquad\quad - (a^{-1}\boldsymbol{u}_R^h, \boldsymbol{\phi}_R) + (p_R^h, \nabla\cdot\boldsymbol{\phi}_R) - \langle g_R, \boldsymbol{n}\cdot\boldsymbol{\phi}_R\rangle_{\Gamma_R} + \langle \xi^h, \boldsymbol{n}\cdot\boldsymbol{\phi}_R\rangle_{\Gamma_I} \\ &\qquad\qquad + (f_R, \zeta_R) - (\nabla\cdot\boldsymbol{u}_R^h, \zeta_R) \\ &\qquad\qquad\quad + \langle \boldsymbol{n}\cdot(\boldsymbol{u}_L^h - \boldsymbol{u}_R^h), \beta\rangle_{\Gamma_I}. \end{aligned} \tag{3.6}$$

We rewrite this as

$$\begin{aligned}(e_{u_L}, \psi_{u_L}) + (e_{p_L}, \psi_{p_L}) + (e_{u_R}, \psi_{u_R}) + (e_{p_R}, \psi_{p_R}) + \langle e_\xi, \psi_\xi \rangle_{\Gamma_I} \\ = (\boldsymbol{R}_{u_L}, \boldsymbol{\phi}_L) + (R_{p_L}, \zeta_L) + (\boldsymbol{R}_{u_R}, \boldsymbol{\phi}_R) + (R_{p_R}, \zeta_R) + \langle R_\xi, \beta \rangle_{\Gamma_I},\end{aligned} \tag{3.7}$$

wherein the residuals are given by

$$\begin{aligned}\boldsymbol{R}_{u_L} = -a^{-1}\boldsymbol{u}_L^h - \nabla p_L^h, \qquad \boldsymbol{R}_{u_R} = -a^{-1}\boldsymbol{u}_R^h - \nabla p_R^h, \\ R_{p_L} = f_L - \nabla \cdot \boldsymbol{u}_L^h, \quad R_{p_R} = f_R - \nabla \cdot \boldsymbol{u}_R^h, \quad R_\xi = \boldsymbol{n} \cdot (\boldsymbol{u}_L^h - \boldsymbol{u}_R^h).\end{aligned}$$

Note that the divergence theorem implies

$$\begin{aligned}(\boldsymbol{R}_{u_L}, \boldsymbol{\phi}_L) &= -(a^{-1}\boldsymbol{u}_L^h, \boldsymbol{\phi}_L) + (p_L^h, \nabla \cdot \boldsymbol{\phi}_L) - \langle p_L^h, \boldsymbol{n} \cdot \boldsymbol{\phi}_L \rangle_{\partial\Omega_L} \\ &= -(a^{-1}\boldsymbol{u}_{h_L}, \boldsymbol{\phi}_L) + (p_L^h, \nabla \cdot \boldsymbol{\phi}_L) - \langle g_L, \boldsymbol{n} \cdot \boldsymbol{\phi}_L \rangle_{\Gamma_L} - \langle \xi^h, \boldsymbol{n} \cdot \boldsymbol{\phi}_L \rangle_{\Gamma_I}.\end{aligned}$$

Also note that $\beta = \zeta_L = \zeta_R$ for the continuous adjoint solution, but β is distinct from ζ_L and ζ_R for the discrete solution.

Next, we use Galerkin orthogonality. We introduce projection operators that map into the finite element space of the discrete forward solution:

$$\begin{aligned}P_L^h : L^2(\Omega_L) \to W_L^h, \quad P_R^h : L^2(\Omega_R) \to W_R^h, \\ \Pi_L^h : L^2(\Omega_L) \to \boldsymbol{V}_L^h, \quad \Pi_R^h : L^2(\Omega_R) \to \boldsymbol{V}_R^h, \quad Z^h : L^2(\Gamma_I) \to \Lambda^h.\end{aligned}$$

The actual choice of projection is immaterial for the estimate. In practice, we employ a combination of restriction and averaging. Without quadrature, Galerkin orthogonality for (2.6) is expressed as

$$(\boldsymbol{R}_{u_L}, \Pi_L^h \boldsymbol{\phi}_L) + (R_{p_L}, P_L^h \zeta_L) + (\boldsymbol{R}_{u_R}, \Pi_R^h \boldsymbol{\phi}_R) + (R_{p_R}, P_R^h \zeta_R) + \langle R_\xi, Z_h \beta \rangle_{\Gamma_I} = 0,$$

and subtracting gives the following result.

THEOREM 3.1 The errors for the mixed finite element method (2.6) without quadrature satisfy

$$\begin{aligned}(e_{p_L}, \psi_{p_L}) + (e_{u_L}, \boldsymbol{\psi}_{u_L}) + (e_{p_R}, \psi_{p_R}) + (e_{u_R}, \boldsymbol{\psi}_{u_R}) + \langle e_\xi, \psi_\xi \rangle_{\Gamma_I} \\ = (\boldsymbol{R}_{u_L}, \boldsymbol{\phi}_L - \Pi_L^h \boldsymbol{\phi}_L) + (R_{p_L}, \zeta_L - P_L^h \zeta_L) \\ + (\boldsymbol{R}_{u_R}, \boldsymbol{\phi}_R - \Pi_R^h \boldsymbol{\phi}_R) + (R_{p_R}, \zeta_R - P_R^h \zeta_R) + \langle R_\xi, \beta - Z_h \beta \rangle_{\Gamma_I},\end{aligned} \tag{3.8}$$

wherein the quantities on the right-hand side are computable provided the true adjoint solution is available.

In practice, we employ a numerical solution of the adjoint problem. To emphasize this, we state the following corollary that involves numerical adjoint quantities.

COROLLARY 3.1 Provided that the projection operators P_L^h, P_R^h, Π_L^h, Π_R^h, and Z_h are bounded in L^2, the errors for the mixed finite element method (2.6) without quadrature can be estimated as

$$\begin{aligned}(e_{p_L}, \psi_{p_L}) + (e_{u_L}, \boldsymbol{\psi}_{u_L}) + (e_{p_R}, \psi_{p_R}) + (e_{u_R}, \boldsymbol{\psi}_{u_R}) + \langle e_\xi, \psi_\xi \rangle_{\Gamma_I} \\ \approx (\boldsymbol{R}_{u_L}, \boldsymbol{\phi}_L^h - \Pi_L^h \boldsymbol{\phi}_L^h) + (R_{p_L}, \zeta_L^h - P_L^h \zeta_L^h) \\ + (\boldsymbol{R}_{u_R}, \boldsymbol{\phi}_R^h - \Pi_R^h \boldsymbol{\phi}_R^h) + (R_{p_R}, \zeta_R^h - P_R^h \zeta_R^h) + \langle R_\xi, \beta^h - Z_h \beta^h \rangle_{\Gamma_I},\end{aligned} \tag{3.9}$$

for numerical solutions $\boldsymbol{\phi}_L \approx \boldsymbol{\phi}_L^h$, $\zeta_L \approx \zeta_L^h$, $\boldsymbol{\phi}_R \approx \boldsymbol{\phi}_R^h$, $\zeta_R \approx \zeta_R^h$, and $\beta \approx \beta^h$. In this approximation, the errors are to be measured in the L^2-norm.

The proof follows from the triangle inequality and the definition of the operator norm. That is, the absolute value of the difference between the right-hand sides of (3.8) and (3.9) is bounded by

$$\begin{aligned}&(1+\|\Pi_L^h\|)\|\boldsymbol{R}_{u_L}\|_2\|\boldsymbol{\phi}_L-\boldsymbol{\phi}_L^h\|_2+(1+\|P_L^h\|)\|R_{p_L}\|_2\|\zeta_L-\zeta_L^h\|_2\\&\quad+(1+\|\Pi_R^h\|)\|\boldsymbol{R}_{u_R}\|_2\|\boldsymbol{\phi}_R-\boldsymbol{\phi}_R^h\|_2+(1+\|P_R^h\|)\|R_{p_R}\|_2\|\zeta_R-\zeta_R^h\|_2\\&\quad+(1+\|Z_h\|)\|R_\xi\|_{2,\Gamma_I}\|\beta-\beta^h\|_{2,\Gamma_I}.\end{aligned}$$

In order to obtain accurate estimates, the numerical adjoint solutions must be sufficiently accurate. Generally this is satisfied by solving the adjoint problems either using a higher order numerical method or using a mesh sufficiently refined from the one used for the forward discretization. In the context of finite volume discretizations, the second approach is generally easier to implement. In our numerical examples we use a finer grid, and the accuracy of this approach is illustrated in section 4.1.

3.2 *Estimate for finite volume methods using geometric coupling*

3.2.1 *The effect of quadrature.* We first derive an estimate for the mixed finite element method (2.6) with quadrature, which can be applied, say, if f, g, and a are continuous. With quadrature, Galerkin orthogonality is expressed as

$$\begin{aligned}&(\boldsymbol{R}_{u_L},\Pi_L^h\boldsymbol{\phi}_L)_Q+(R_{p_L},P_L^h\zeta_L)_Q\\&\quad+(\boldsymbol{R}_{u_R},\Pi_R^h\boldsymbol{\phi}_R)_Q+(R_{p_R},P_R^h\zeta_R)_Q+\langle R_\xi,Z_h\beta\rangle_{Q,\Gamma_I}=0,\end{aligned}$$

where we use the subscript Q to denote the approximate inner product using quadrature. It is important to distinguish residuals associated with approximating the solution spaces using finite dimensional polynomial spaces from residuals associated with approximating the integrals defining the variational formulation. We rewrite Galerkin orthogonality as

$$\begin{aligned}&(\boldsymbol{R}_{u_L},\Pi_L^h\boldsymbol{\phi}_L)+(R_{p_L},P_L^h\zeta_L)+(\boldsymbol{R}_{u_R},\Pi_R^h\boldsymbol{\phi}_R)+(R_{p_R},P_R^h\zeta_R)+\langle R_\xi,Z_h\beta\rangle_{\Gamma_I}\\&\quad-QE_{u_L}(\Pi_L^h\boldsymbol{\phi}_L)-QE_{p_L}(P_L^h\zeta_L)-QE_{u_R}(\Pi_R^h\boldsymbol{\phi}_R)-QE_{p_R}(P_R^h\zeta_R)-QE_\xi(Z^h\beta)=0,\end{aligned}$$

with

$$\begin{aligned}QE_{u_L}(\Pi_L^h\boldsymbol{\phi}_L)&=(\boldsymbol{R}_{u_L},\Pi_L^h\boldsymbol{\phi}_L)-(\boldsymbol{R}_{u_L},\Pi_L^h\boldsymbol{\phi}_L)_Q,\\QE_{p_L}(P_L^h\zeta_L)&=(R_{p_L},P_L^h\zeta_L)-(R_{p_L},P_L^h\zeta_L)_Q,\\QE_{u_R}(\Pi_R^h\boldsymbol{\phi}_R)&=(\boldsymbol{R}_{u_R},\Pi_R^h\boldsymbol{\phi}_R)-(\boldsymbol{R}_{u_R},\Pi_R^h\boldsymbol{\phi}_R)_Q,\\QE_{p_R}(P_R^h\zeta_R)&=(R_{p_R},P_R^h\zeta_R)-(R_{p_R},P_R^h\zeta_R)_Q,\\QE_\xi(Z^h\beta)&=\langle R_\xi,Z_h\beta\rangle_{\Gamma_I}-\langle R_\xi,Z_h\beta\rangle_{Q,\Gamma_I}.\end{aligned}$$

This gives the following a-posteriori estimate for the mixed finite element method with quadrature.

THEOREM 3.2 If f, g, and a are continuous, then the errors for the mixed finite element method (2.6) with quadrature satisfy

$$\begin{aligned}&(e_{p_L},\psi_{p_L})+(e_{u_L},\boldsymbol{\psi}_{u_L})+(e_{p_R},\psi_{p_R})+(e_{u_R},\boldsymbol{\psi}_{u_R})+\langle e_\xi,\psi_\xi\rangle_{\Gamma_I}\\&\quad=(\boldsymbol{R}_{u_L},\boldsymbol{\phi}_L-\Pi_L^h\boldsymbol{\phi}_L)+(R_{p_L},\zeta_L-P_L^h\zeta_L)\\&\qquad+(\boldsymbol{R}_{u_R},\boldsymbol{\phi}_R-\Pi_R^h\boldsymbol{\phi}_R)+(R_{p_R},\zeta_R-P_R^h\zeta_R)+\langle R_\xi,\beta-Z_h\beta\rangle_{\Gamma_I}\\&\qquad+QE_{u_L}(\Pi_L^h\boldsymbol{\phi}_L)+QE_{p_L}(P_L^h\zeta_L)+QE_{u_R}(\Pi_R^h\boldsymbol{\phi}_R)+QE_{p_R}(P_R^h\zeta_R)+QE_\xi(Z^h\beta).\end{aligned}\tag{3.10}$$

Note that in the case of using the RT0 finite element space and the midpoint-trapezoidal quadrature rules discussed above, the mixed finite element method reduces to the finite volume method (Russell & Wheeler, 1983; Weiser & Wheeler, 1988; Estep *et al.*, 2009a), and some of the quadrature error terms are zero. These terms are included for generality, so that (3.10) is valid for other combinations of finite element spaces and quadratures.

Note that in practice, we implement the obvious analog of Corollary 3.1, which now requires sufficient smoothness of the solution to obtain sufficiently accurate quadrature approximations.

3.2.2 *The effect of geometric coupling.* For the geometric coupling (2.8), the Galerkin orthogonality becomes

$$(\boldsymbol{R}_{u_L}, \Pi_L^h \boldsymbol{\phi}_L)_Q - \langle P_{R\to L}(p_R^h) - \xi^h, \boldsymbol{n}\cdot\Pi_L^h\boldsymbol{\phi}_L\rangle_{Q,\Gamma_I} + (R_{p_L}, P_L^h\zeta_L)_Q$$
$$+ (\boldsymbol{R}_{u_R}, \Pi_R^h\boldsymbol{\phi}_R)_Q + (R_{p_R}, P_R^h\zeta_R)_Q + \langle R_\xi, Z_h\beta\rangle_{Q,\Gamma_I} - \langle \boldsymbol{n}\cdot\boldsymbol{u}_L^h - P_{L\to R}(p_L^h), Z^h\beta\rangle_{Q,\Gamma_I} = 0.$$

Defining

$$\mathcal{QE}_{u_L}(\Pi_L^h\boldsymbol{\phi}_L) = (\boldsymbol{R}_{u_L}, \Pi_L^h\boldsymbol{\phi}_L) - (\boldsymbol{R}_{u_L}, \Pi_L^h\boldsymbol{\phi}_L)_Q$$
$$+ \langle P_{R\to L}(p_R^h) - \xi^h, \Pi_L^h\boldsymbol{\phi}_L\rangle_{Q,\Gamma_I} - \langle P_{R\to L}(p_R^h) - \xi^h, \boldsymbol{n}\cdot\Pi_L^h\boldsymbol{\phi}_L\rangle_{\Gamma_I},$$
$$\mathcal{QE}_\xi(Z^h\phi_1) = \langle R_\xi, Z_h\beta\rangle_{\Gamma_I} - \langle R_\xi, Z_h\beta\rangle_{Q,\Gamma_I}$$
$$+ \langle \boldsymbol{n}_L\cdot\boldsymbol{u}_L^h - P_{L\to R}(p_L^h), Z^h\beta\rangle_{Q,\Gamma_I} - \langle \boldsymbol{n}\cdot\boldsymbol{u}_L^h - P_{L\to R}(p_L^h), Z^h\beta\rangle_{\Gamma_I},$$

and arguing as above gives the following result.

THEOREM 3.3 If f, g, and a are continuous, then the error for the mixed geometric finite volume method (2.8) satisfies

$$(e_{p_L}, \psi_{p_L}) + (e_{u_L}, \boldsymbol{\psi}_{u_L}) + \langle e_{p_R}, \psi_{p_R}\rangle + \langle e_{u_R}, \boldsymbol{\psi}_{u_R}\rangle + \langle e_\xi, \psi_\xi\rangle_{\Gamma_I} \tag{3.11}$$
$$= \langle \boldsymbol{R}_{u_L}, \boldsymbol{\phi}_L - \Pi_L^h\boldsymbol{\phi}_L\rangle + \langle P_{R\to L}(p_R^h) - \xi^h, \boldsymbol{n}\cdot\Pi_L^h\boldsymbol{\phi}_L\rangle_{\Gamma_I} + (R_{p_L}, \zeta_L - P_L^h\zeta_L)$$
$$+ (\boldsymbol{R}_{u_R}, \boldsymbol{\phi}_R - \Pi_R^h\boldsymbol{\phi}_R) + (R_{p_R}, \zeta_R - P_R^h\zeta_R)$$
$$+ \langle R_\xi, \beta - Z_h\beta\rangle_{\Gamma_I} + \langle \boldsymbol{n}\cdot\boldsymbol{u}_L^h - P_{L\to R}(p_L^h), Z^h\beta\rangle_{\Gamma_I}$$
$$+ \mathcal{QE}_{u_L}(\Pi_L^h\boldsymbol{\phi}_L) + QE_{p_L}(P_L^h\zeta_L) + QE_{u_R}(\Pi_R^h\boldsymbol{\phi}_R) + QE_{p_R}(P_R^h\zeta_R) + \mathcal{QE}_\xi(Z^h\beta).$$

Note that in practice, we implement the obvious analog of Corollary 3.1, assuming again sufficient smoothness of the solution to obtain sufficiently accurate quadrature approximations.

4. Numerical investigations

In this section, we use the a-posteriori error estimates to investigate in detail the accuracy of the two approaches to coupling. For all of the investigations, the coarser subdomain Ω_L is given by $x \in [-1, 1]$ and $y \in [-2, 0]$, the finer subdomain Ω_R is given by $x \in [-1, 1]$ and $y \in [0, 2]$ (see Fig. 1), and the interface Γ_I is located along $y = 0$. (Note that here the bottom subdomain is considered as being "left" and the top one is "right," in conformance to our convention as to the finer subdomain.) The grids are reported as $n_L \times m_L$ for the left domain and $n_R \times m_R$ for the right domain, where $n_{(\cdot)}$ corresponds to the number of cells in the x-direction (which is also the number of cells along the interface), and $m_{(\cdot)}$

corresponds to the number of cells in the y-direction. The boundary conditions for all tests are Dirichlet. To avoid issues arising from iterative solution of the discrete system, we employ direct methods to find the approximate solution to within machine precision.

The quantity of interest being sought is specified by giving the adjoint problem data ψ_{u_x}, ψ_{u_y}, ψ_p, and ψ_ξ. The adjoint problem is solved using the same RT0 mixed finite element method, but on a grid that is significantly finer than that of the forward problem, so that the discretization error associated with the adjoint solution has no significant effect on the results.

The functions chosen for the source, diffusivity, and adjoint data are either constants or Gaussian functions of the form

$$\frac{ae^{-(y-b)^2}}{\sqrt{2c^2}} + K,$$

which gives a localized "ridge" centered at $y = b$. In the case of the adjoint data, the Gaussian or constant function being used is normalized so that the area under ψ is equal to one. The parameter K is non zero only in the case of diffusivity, where this constant is added to the Gaussian to prevent the diffusivity from approaching zero anywhere in the domain.

In the tests, we report values for the terms in (3.10) and (3.11) that are non zero. For both the mixed finite element and geometric finite volume methods the following five terms are included:

$$\begin{aligned}
MFE_1 \quad &\text{or} \quad GFV_1 = (\boldsymbol{R}_{u_L}, \boldsymbol{\phi}_L^h - \Pi_L^h \boldsymbol{\phi}_L^h), \\
MFE_2 \quad &\text{or} \quad GFV_2 = (\boldsymbol{R}_{u_R}, \boldsymbol{\phi}_R^h - \Pi_R^h \boldsymbol{\phi}_R^h), \\
MFE_3 \quad &\text{or} \quad GFV_3 = (R_{p_L}, \zeta_L^h - P_L^h \zeta_L^h), \\
MFE_4 \quad &\text{or} \quad GFV_4 = (R_{p_R}, \zeta_R^h - P_R^h \zeta_R^h), \\
MFE_5 \quad &\text{or} \quad GFV_5 = \langle R_\xi, \beta^h - Z_h \beta^h \rangle_{\Gamma_I}.
\end{aligned}$$

In the geometric finite volume case, we add two additional terms relating to the geometric projections and two additional quadrature terms:

$$\begin{aligned}
GFV_6 &= \langle P_{R\to L}(p_R^h) - \xi^h, \boldsymbol{n} \cdot \Pi_L^h \boldsymbol{\phi}_L^h \rangle_{\Gamma_I}, \\
GFV_7 &= \langle \boldsymbol{n} \cdot \boldsymbol{u}_L^h - P_{L\to R}(p_L^h), Z^h \beta^h \rangle_{\Gamma_I}, \\
GFV_8 &= \mathcal{QE}_{u_L}(\Pi_L^h \boldsymbol{\phi}_L^h), \\
GFV_9 &= QE_{u_R}(\Pi_R^h \boldsymbol{\phi}_R^h).
\end{aligned}$$

We note that the first five expressions, common to both MFE and GFV, are often similar in size. As a gross measure of the effect of geometric progression and of the use of quadrature, we also report the two ratios

$$\text{ratio}_{\text{proj}} = \frac{\sum_{i=6}^{7} |\, GFV_i \,|}{\sum_{i=1}^{5} |\, GFV_i \,|}, \qquad \text{ratio}_{\text{quad}} = \frac{\sum_{i=8}^{9} |\, GFV_i \,|}{\sum_{i=1}^{5} |\, GFV_i \,|}.$$

4.1 *Verification of a-posteriori estimate accuracy*

We begin with a problem for which we have manufactured the known solution

$$p(x,y) = \cos\left(\frac{\pi x}{2}\right) \cos\left(\frac{\pi y}{4}\right). \tag{4.1}$$

The diffusivity a is equal to one everywhere. The other solution components, the source term f, and the boundary values g for the problem follow from (4.1). Since we know the true solution, we can compute the exact error terms (e, ψ) on the left in (3.10) and (3.11) directly and then compare to estimates of the quantities on the right computed using a numerical solution to the adjoint problem. In this situation, the most important issue for the accuracy of the estimates is the accuracy of the approximate adjoint solutions. As the grid for the adjoint problem is refined, the estimates become more accurate. That is, using the approximation to the adjoint problem, the estimated quantities $\sum MFE_i$ or $\sum GFV_i$ becomes closer to their true value, the error in the quantity of interest MFE $\sum(e, \psi)$ or GFV $\sum(e, \psi)$. Tables 1 and 2 show this using coarse and fine forward solutions.

Table 1. The forward problem with solution (4.1) is run at 10×10 next to 16×16. The adjoint problem is run at several grids to show how the sum of terms approaches the direct calculation of (e, ψ). The adjoint data components ψ_u and ψ_p are constant everywhere and $\psi_\xi = 0$.

adj. grid	MFE $\sum(e, \psi)$	$\sum MFE_i$	ratio	GFV $\sum(e, \psi)$	$\sum GFV_i$	ratio
20x20 : 32x32	$1.96E-3$	$1.47E-3$	.749	$-1.00E-3$	$-1.50E-3$	1.49
40x40 : 64x64	$1.96E-3$	$1.84E-3$	.937	$-1.00E-3$	$-1.13E-3$	1.12
80x80 : 128x128	$1.96E-3$	$1.93E-3$	.984	$-1.00E-3$	$-1.03E-3$	1.03
160x160 : 256x256	$1.96E-3$	$1.96E-3$	.996	$-1.00E-3$	$-1.01E-3$	1.01

Table 2. The forward problem with solution (4.1) is run at 40×40 next to 64×64. The adjoint problem is run at several grids to show how the sum of terms approaches the direct calculation of (e, ψ). The adjoint data components ψ_u and ψ_p are constant everywhere and $\psi_\xi = 0$.

adj. grid	MFE $\sum(e, \psi)$	$\sum MFE_i$	ratio	GFV $\sum(e, \psi)$	$\sum GFV_i$	ratio
80x80 : 128x128	$1.23E-4$	$9.23E-5$	.750	$-7.00E-5$	$-1.01E-4$	1.44
160x160 : 256x256	$1.23E-4$	$1.15E-4$	.937	$-7.00E-5$	$-7.77E-5$	1.11

4.2 *Convergence*

To compare the accuracy of the various approximations, we use the 2-norms

$$\|e_p\|_2 = \sqrt{\int_\Omega (p - p^h)^2}, \quad \|e_{u_x}\|_2 = \sqrt{\int_\Omega (u_x - u_x^h)^2},$$
$$\|e_{u_y}\|_2 = \sqrt{\int_\Omega (u_y - u_y^h)^2}, \quad \|e_\xi\|_2 = \sqrt{\int_\Gamma (\xi - \xi^h)^2}.$$

We use the manufactured solution from the previous section ($a = 1$ and p is given by (4.1)). We compare the 2-norm errors of the finite element and geometric finite volume methods on a sequence of grids in order to asses the convergence rate. The coarsest grid is 10×10 next to 16×16, and the number of cells in each dimension is doubled with each refinement.

The results in Tables 3–6 show that the convergence rate for the geometric finite volume deteriorates for the u_x, u_y, and ξ components when the number of cells along the fine side of the interface is not an integer multiple of the number of cells along the coarse side of the interface. When the test is repeated with a grid starting at 8×8 next to 16×16, the convergence rates for the two methods are equal. The first order convergence of p and u for the MFE is to be expected (Arbogast *et al.*, 2000).

Table 3. Convergence of solution component p, indicating a rate of $O(h)$.

grid	MFE $\|e_p\|$	MFE ratio	GFV $\|e_p\|$	GFV ratio
10x10 : 16x16	1.20E-01	N/A	1.20E-01	N/A
20x20 : 32x32	5.98E-02	2.00	5.98E-02	2.00
40x40 : 64x64	2.99E-02	2.00	2.99E-02	2.00
80x80 : 128x128	1.49E-02	2.00	1.49E-02	2.00
160x160 : 256x256	7.47E-03	2.00	7.47E-03	2.00

Table 4. Convergence of solution component u_x, indicating a rate of about $O(h)$.

grid	MFE $\|e_{u_x}\|$	MFE ratio	GFV $\|e_{u_x}\|$	GFV ratio
10x10 : 16x16	8.49E-02	N/A	8.63E-02	N/A
20x20 : 32x32	4.21E-02	2.02	4.26E-02	2.02
40x40 : 64x64	2.10E-02	2.00	2.14E-02	1.99
80x80 : 128x128	1.05E-02	2.00	1.09E-02	1.97
160x160 : 256x256	5.25E-03	2.00	5.63E-03	1.93

Table 5. Convergence of solution component u_y, indicating a rate of $O(h)$ for MFE but less for GFV.

grid	MFE $\|e_{u_y}\|$	MFE ratio	GFV $\|e_{u_y}\|$	GFV ratio
10x10 : 16x16	8.41E-02	N/A	8.59E-02	N/A
20x20 : 32x32	4.20E-02	2.00	4.39E-02	1.96
40x40 : 64x64	2.10E-02	2.00	2.29E-02	1.92
80x80 : 128x128	1.05E-02	2.00	1.23E-02	1.86
160x160 : 256x256	5.25E-03	2.00	6.94E-03	1.77

Table 6. Convergence of solution component ξ, indicating a rate of $O(h^2)$ for MFE but only $O(h)$ for GFV.

grid	MFE $\|e_\xi\|$	MFE ratio	GFV $\|e_\xi\|$	GFV ratio
10x10 : 16x16	7.53e-03	N/A	6.11e-03	N/A
20x20 : 32x32	1.89e-03	3.99	1.79e-03	3.42
40x40 : 64x64	4.72e-04	4.00	6.37e-04	2.80
80x80 : 128x128	1.18e-04	4.00	2.77e-04	2.30
160x160 : 256x256	2.95e-05	4.00	1.33e-04	2.09

4.3 *Test Case 1*

In the next problem, we explore accuracy for a solution that is not changing rapidly near the interface. We find that the use of geometric projections does not lead to significant effects on accuracy. We let the diffusivity a be one everywhere and use the manufactured solution given by (4.1). The grid for the forward problem is 20×20 next to 32×32. The adjoint grid is 80×80 next to 128×128, and the adjoint data is a nonzero constant for ψ_{u_x}, ψ_{u_y}, and ψ_p, while $\psi_\xi = 0$.

We list the error contributions in Table 7. For the geometric approach, we list results for both constant and linear extrapolation. The results show that the projection error for linear extrapolation is only about one quarter of the residual error, while the projection error for constant extrapolation is much

larger. Fig. 5 shows the solution components for the finite element case. The geometric finite volume solutions are very similar. Fig. 6 shows the adjoint solution components.

Table 7. Error terms for Case 1. The forward grid is 20×20 next to 32×32. The adjoint grid is 80×80 next to 128×128.

term		*MFE*	*GFV(linear)*	*GFV(constant)*
1	$(\boldsymbol{R}_{u_L}, \boldsymbol{\phi}_L^h - \Pi_L^h \boldsymbol{\phi}_L^h)$	$-1.6E{-}4$	$-1.6E{-}4$	$-1.5E{-}4$
2	$(\boldsymbol{R}_{u_R}, \boldsymbol{\phi}_R^h - \Pi_R^h \boldsymbol{\phi}_R^h)$	$-6.1E{-}5$	$-6.1E{-}5$	$-6.1E{-}5$
3	$(R_{p_L}, \zeta_L^h - P_L^h \zeta_L^h)$	$4.9E{-}4$	$4.9E{-}4$	$4.9E{-}4$
4	$(R_{p_R}, \zeta_R^h - P_R^h \zeta_R^h)$	$1.9E{-}4$	$1.9E{-}4$	$1.9E{-}4$
5	$(R_\xi, \beta^h - Z_h \beta^h)_{\Gamma_I}$	$4.2E{-}8$	$-1.3E{-}6$	$1.0E{-}5$
6	$(P_{R\to L}(p_R^h) - \xi^h, \boldsymbol{n} \cdot \Pi_L^h \boldsymbol{\phi}_L^h)_{\Gamma_I}$	N/A	$2.0E{-}4$	$1.7E{-}4$
7	$(\boldsymbol{n} \cdot \boldsymbol{u}_L^h - P_{L\to R}(p_L^h), Z^h \beta^h)_{\Gamma_I}$	N/A	$2.2E{-}5$	$3.9E{-}3$
8	$\mathcal{QE}_{u_L}(\Pi_L^h \boldsymbol{\phi}_L^h)$	N/A	$-7.1E{-}4$	$-7.0E{-}4$
9	$\mathcal{QE}_{u_R}(\Pi_R^h \boldsymbol{\phi}_R^h)$	N/A	$-2.8E{-}4$	$2.7E{-}4$
total		$4.6E{-}4$	$-3.0E{-}4$	$3.6E{-}3$
$\text{ratio}_{\text{proj}}$		N/A	.25	4.5
$\text{ratio}_{\text{quad}}$		N/A	1.1	1.1

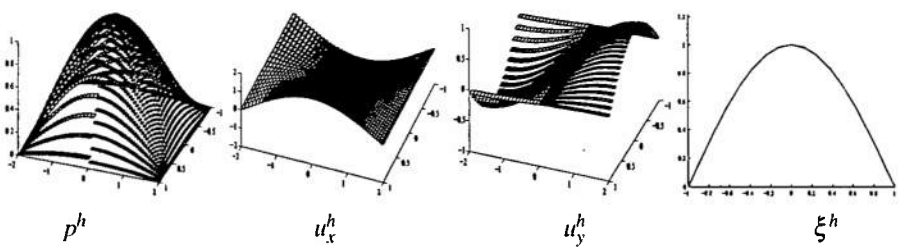


FIG. 5. Finite element solution components for Case 1.

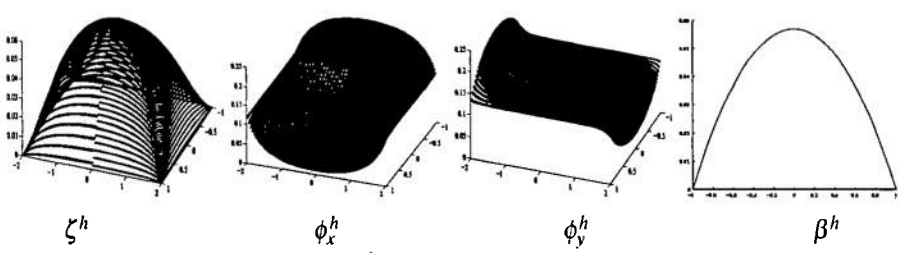


FIG. 6. Adjoint solution components for Case 1. Shown are plots for an adjoint solution using a 40×40 grid next to 64×64 grid. A solution on a finer grid is used to compute the estimates.

4.4 *Test Case 2*

The next test problem presents a more difficult solution for which the geometric projection error is by far the largest source of error. The grid is 40×40 next to 64×64 and the boundary conditions are $g = 0$

on both subdomains. Fig. 7 shows profiles of the source and diffusivity, while Fig. 8 shows the adjoint data. [1]

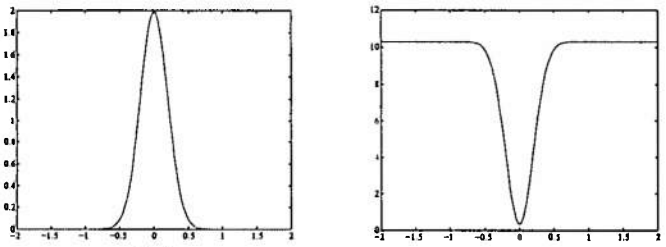


FIG. 7. Source f (left) and diffusivity a (right) profiles for Test Case 2. The plots are shown in one dimension since the source and diffusivity have no variation in the x-direction.

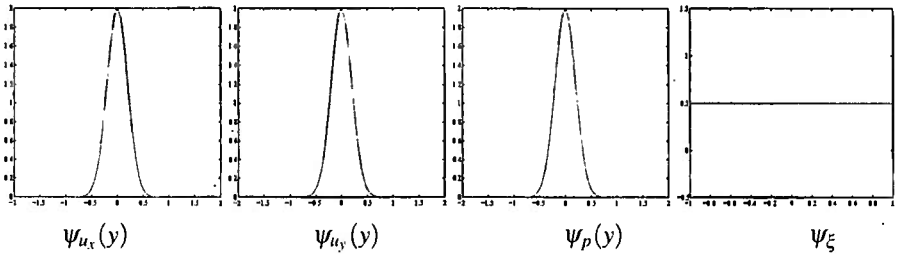


FIG. 8. Adjoint data profiles for Case 2. The plots of ψ_{u_x}, ψ_{u_y}, and ψ_p are shown in one dimension because they have no variation in the x-direction.

Because the source is large but the diffusivity is small along the interface, the solution changes rapidly near this region. This leads to relatively large errors near the interface for the geometric finite volume method. When the adjoint data is concentrated near the interface, the relative size of these errors is revealed. Table 8 lists the error terms. For this particular example problem, and this particular error measure, the error due to geometric projection is nearly eighty times the total error associated with the residuals. Fig. 9 shows the solution components for the finite element case, Fig. 10 shows the solution components for the geometric finite volume case, and Fig. 11 shows the adjoint solution.

[1]The shapes in Fig. 7 and 8 are based on a normalized gaussian of the form $\frac{ae^{(x-b)^2}}{2c^2}$. The parameter a is for normalization and is set to $a = \frac{1}{c\sqrt{2\pi}}$. The parameter b determiness the location of the peak and is set to zero to coincide with the interaface. The parameter c determines the width of the peak and is set to $c = .2$. In the case of diffusivity the dip is produced using the function $10.3 - 10 * (\frac{ae^{(x-b)^2}}{2c^2})$.

Table 8. Error terms for Case 2. The forward grid is 40×40 next to 64×64. The adjoint grid is 160×160 next to 256×256.

term		*MFE*	*GFV(linear)*	*GFV(constant)*
1	$(\boldsymbol{R}_{u_L}, \boldsymbol{\phi}_L^h - \Pi_L^h \boldsymbol{\phi}_L^h)$	$1.6E-5$	$1.9E-5$	$2.4E-5$
2	$(\boldsymbol{R}_{u_R}, \boldsymbol{\phi}_R^h - \Pi_R^h \boldsymbol{\phi}_R^h)$	$-2.6E-5$	$-2.6E-5$	$-2.5E-5$
3	$(R_{p_L}, \zeta_L^h - P_L^h \zeta_L^h)$	$-3.1E-5$	$-3.1E-5$	$-3.1E-5$
4	$(R_{p_R}, \zeta_R^h - P_R^h \zeta_R^h)$	$4.6E-5$	$4.6E-5$	$4.6E-5$
5	$\langle R_\xi, \beta^h - Z_h \beta^h \rangle_{\Gamma_I}$	$4.8E-8$	$9.4E-6$	$2.6E-5$
6	$\langle P_{R \to L}(p_R^h) - \xi^h, \boldsymbol{n} \cdot \Pi_L^h \boldsymbol{\phi}_L^h \rangle_{\Gamma_I}$	N/A	$2.3E-3$	$2.7E-3$
7	$\langle \boldsymbol{n} \cdot \boldsymbol{u}_L^h - P_{L \to R}(p_L^h), Z^h \beta^h \rangle_{\Gamma_I}$	N/A	$9.0E-4$	$8.4E-3$
8	$QE_{u_L}(\Pi_L^h \boldsymbol{\phi}_L^h)$	N/A	$1.2E-3$	$1.2E-3$
9	$QE_{u_L}(\Pi_L^h \boldsymbol{\phi}_L^h)$	N/A	$-2.4E-4$	$-2.5E-4$
total		$5.1E-6$	$4.2E-3$	$1.2E-2$
$\text{ratio}_{\text{proj}}$		N/A	25	73
$\text{ratio}_{\text{quad}}$		N/A	11	9.7

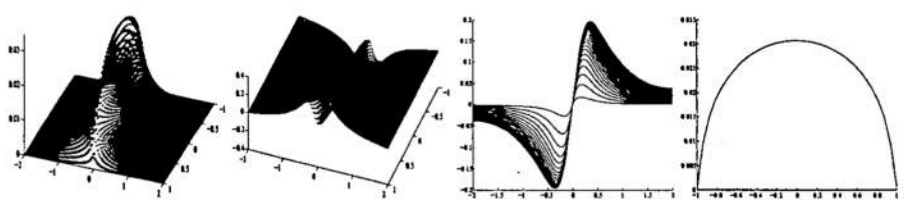


FIG. 9. Finite element solution components for Case 2. Zooming in reveals that u_y^h is smooth and continuous across the interface.

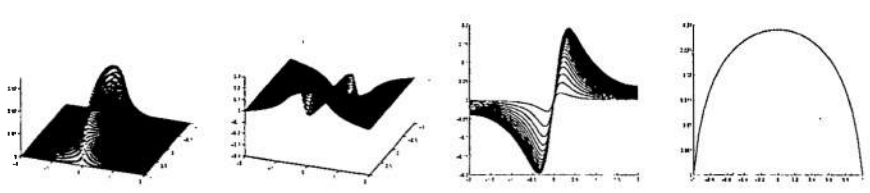


FIG. 10. Geometric finite volume solution components for Case 2. Zooming in reveals that u_y^h is discontinuous across the interface.

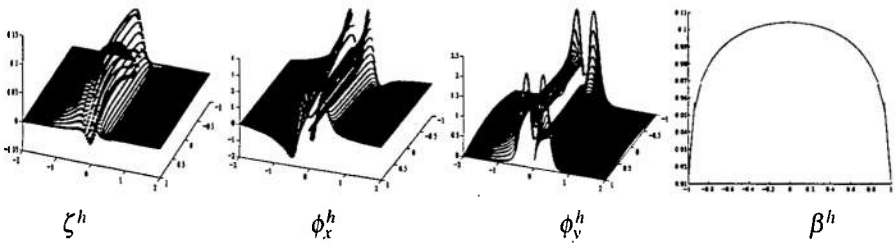


FIG. 11. Adjoint solution components for Case 2. Shown are plots for an adjoint solution using a 40×40 grid next to 64×64 grid. A solution on a finer grid is used to compute the estimates.

4.5 *Test Case 3*

In our final example, we examine a problem that places only one cell in the x-direction in one of the subdomains. Such a grid is only appropriate if the solution in that subdomain is essentially one dimensional, and varies only parallel to the interface. This situation arises in core-edge coupling in a tokamak fusion reactor.

We construct a problem with a solution that is very nearly one dimensional in one subdomain, and contains variation in the second dimension well away from the interface. The pressure component of the solution is

$$p(x,y) = \cos\left(\frac{\pi(y+2)}{8}\right) + 0.3\sin(\pi x)\left[\frac{1-\tanh\left(2(1.5-y)\right)}{2}\right]. \tag{4.2}$$

The grid is 1×32 next to 32×32 and the boundary conditions are provided by evaluating the known solution at the outer domain boundaries. The source for the problem is computed by substituting the chosen solution into the PDE. The diffusivity a is one everywhere. The adjoint data is concentrated in the finer subdomain, and is shown in Fig. 12.

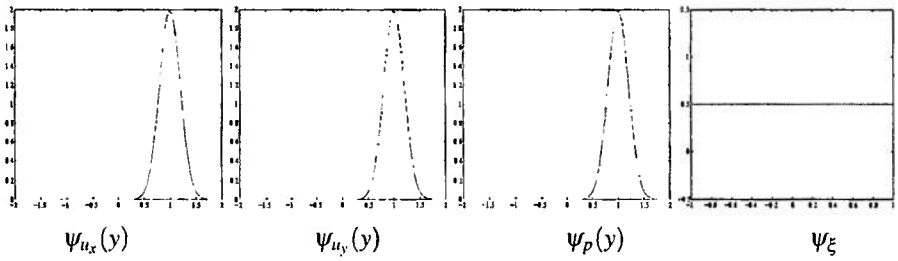


FIG. 12. Adjoint data profiles for Case 3. The plots of ψ_{u_x}, ψ_{u_y}, and ψ_p are shown in one dimension because they have no variation in the x-direction, and ψ_ξ is a one dimensional function defined on the interface.

Table 9 lists the error terms. For this example problem, the contribution due to geometric projection with linear extrapolation is approximately ten times the total contribution associated with the residuals, despite the fact that the solution is changing slowly near the interface. The projection contribution is much larger if constant extrapolation is used. Fig. 13 shows the solution components for the finite element case, Fig. 14 shows the solution components for the geometric finite volume case, and Fig. 15 shows the adjoint solution components.

Table 9. Error terms for Case 3. The forward grid is 1×32 next to 32×32. The adjoint grid is 128×128 next to 128×128.

	term	MFE	GFV (linear)	GFV (constant)
1	$(\boldsymbol{R}_{u_L}, \boldsymbol{\phi}_L^h - \Pi_L^h \boldsymbol{\phi}_L^h)$	$3.9E{-}9$	$6.8E{-}7$	$-1.5E{-}5$
2	$(\boldsymbol{R}_{u_R}, \boldsymbol{\phi}_R^h - \Pi_R^h \boldsymbol{\phi}_R^h)$	$1.8E{-}5$	$1.8E{-}5$	$1.8E{-}5$
3	$(R_{p_L}, \zeta_L^h - P_L^h \zeta_L^h)$	$-4.0E{-}6$	$-4.0E{-}6$	$-4.0E{-}6$
4	$(R_{p_R}, \zeta_R^h - P_R^h \zeta_R^h)$	$7.2E{-}6$	$7.2E{-}6$	$7.2E{-}6$
5	$(R_\xi, \beta^h - Z_h \beta^h)_{\Gamma_I}$	0	$-3.8E{-}7$	$1.7E{-}5$
6	$\langle P_{R\to L}(p_R^h) - \xi^h, \boldsymbol{n} \cdot \Pi_L^h \boldsymbol{\phi}_L^h \rangle_{\Gamma_I}$	N/A	$1.5E{-}4$	$-6.4E{-}3$
7	$\langle \boldsymbol{n} \cdot \boldsymbol{u}_L^h - P_{L\to R}(p_L^h), Z^h \beta^h \rangle_{\Gamma_I}$	N/A	$-6.8E{-}5$	$3.1E{-}3$
8	$\mathcal{QE}_{u_L}(\Pi_L^h \boldsymbol{\phi}_L^h)$	N/A	$-2.4E{-}5$	$2.5E{-}3$
9	$\mathcal{QE}_{u_R}(\Pi_R^h \boldsymbol{\phi}_R^h)$	N/A	$2.0E{-}5$	$1.9E{-}5$
total		$2.1E{-}5$	$1.0E{-}4$	$-8.0E{-}4$
$\text{ratio}_{\text{proj}}$		N/A	7.3	155
$\text{ratio}_{\text{quad}}$		N/A	1.5	41

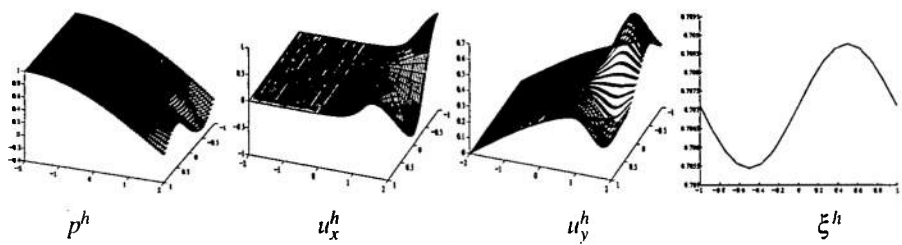


FIG. 13. Finite element solution components for Case 3.

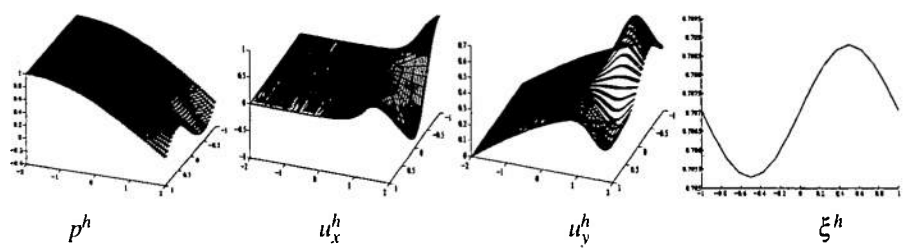


FIG. 14. Geometric finite volume solution components for Case 3 computed using linear extrapolation.

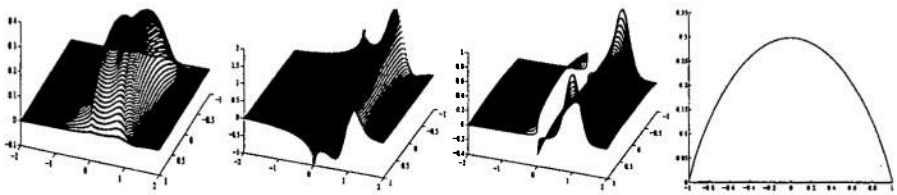


FIG. 15. Adjoint solution components for Case 3. These plots are the adjoint solution using 64×64 next to 64×64 meshes. The estimates were computing using a finer grid.

5. Iterative solvers and coupling strategies

In practice, iterative solution of the coupled system is often employed. The specific choice of solution method is often constrained by certain computational logistics, such as the state of existing codes and data structures. We briefly discuss some aspects of iterative solution. The primary goal is to show that iterative solution strategies applied to systems like (2.10) can also be applied to systems like (2.7) without large changes to the computational structure. We do not discuss the convergence of iterative solvers.

5.1 *Iteration on the primary variable*

A common iterative technique for the geometric finite volume method (2.10) is to start with an initial guess (p_L^0, p_R^0) and proceed with the iteration

$$\begin{bmatrix} A_L & 0 \\ 0 & A_R \end{bmatrix} \begin{bmatrix} p_L^{i+1} \\ p_R^{i+1} \end{bmatrix} = \begin{bmatrix} F_L \\ F_R \end{bmatrix} - \begin{bmatrix} 0 & C_D \\ C_N & 0 \end{bmatrix} \begin{bmatrix} p_L^i \\ p_R^i \end{bmatrix}, \quad i = 0, 1, 2, \ldots. \tag{5.1}$$

This iteration requires only the inversion of A_L and A_R, that is, only single domain component solves. The application of C_D and C_N can be viewed as the coupling strategy, in which information is swapped between the subdomains.

It is possible to use an iteration of this type on the finite element system (2.7) as well. We must first reduce to a system in p by a preprocessing procedure. We first eliminate u_L and u_R, which results in

$$\begin{bmatrix} B_L^T M_L^{-1} B_L & 0 & -B_L^T M_L^{-1} C_L \\ 0 & B_R^T M_R^{-1} B_R & -B_R^T M_R^{-1} C_R \\ C_L^T M_L^{-1} B_L & C_R^T M_R^{-1} B_R & -(C_L^T M_L^{-1} C_L + C_R^T M_R^{-1} C_R) \end{bmatrix} \begin{bmatrix} p_L \\ p_R \\ \xi \end{bmatrix} = \begin{bmatrix} F_L + B_L^T M_L^{-1} D_L \\ F_R + B_R^T M_R^{-1} D_R \\ C_L^T M_L^{-1} D_L + C_R^T M_R^{-1} D_R \end{bmatrix}, \tag{5.2}$$

which we write succinctly as

$$\begin{bmatrix} G_L & 0 & -H_L \\ 0 & G_R & -H_R \\ H_L^T & H_R^T & -(K_L + K_R) \end{bmatrix} \begin{bmatrix} p_L \\ p_R \\ \xi \end{bmatrix} = \begin{bmatrix} R_L \\ R_R \\ S_L + S_R \end{bmatrix}. \tag{5.3}$$

We then eliminate ξ to obtain

$$\begin{bmatrix} G_L - H_L(K_L+K_R)^{-1}H_L^T & -H_L(K_L+K_R)^{-1}H_R^T \\ -H_R(K_L+K_R)^{-1}H_L^T & G_R - H_R(K_L+K_R)^{-1}H_R^T \end{bmatrix} \begin{bmatrix} p_L \\ p_R \end{bmatrix} = \begin{bmatrix} R_L - H_L(K_L+K_R)^{-1}(S_L+S_R) \\ R_R - H_R(K_L+K_R)^{-1}(S_L+S_R) \end{bmatrix}. \quad (5.4)$$

System (5.4) has the same structure as (2.10), so an iteration analogous to (5.1) can be applied. The stencil within the diagonal blocks of (5.4) is very close, but not identical, to the stencil of a single domain discretization. The difference occurs only in the stencil corresponding to cells touching the interface.

In some cases, e.g., the use of black box single domain solvers, it is necessary to construct a system in which the diagonal blocks correspond exactly to single domain discretizations. If this is the case, the strategy of "discretization consistent interface conditions" provides a partial solution. In this strategy, the diagonal blocks are single domain discretizations, just as in (2.10). The off diagonal blocks are populated by writing down both the Dirichlet and Neumann boundary condition equations for every cell touching the interface, rearranging those equations to isolate the boundary value terms and setting those terms equal to each other across the interface. If the cell ratio along the interface is integer, such as 4 next to 8, the resulting system is algebraically equivalent to (5.4). If the cell ratio is not an integer ratio, such as 5 next to 8, the equality of boundary value terms across the interface can only be enforced approximately, and the resulting system is not exactly equivalent to (5.4). While a complete discussion of the implementation of discretization consistent interface conditions is beyond the scope of this paper, it is worth consideration as an alternative to the full mortar method in cases where the computational structure is constrained by black box single domain solvers in combination with iteration on the primary variables. The concept of discretization consistent interface conditions is similar to strategies employed in Farhat *et al.* (1998) and Edwards & Rogers (1998). We should remark that the former paper recommended against mortar methods for the fluid-structure interaction problem, due to the lack of theory on optimal convergence and a need to invert a large interface matrix. However, for the problem considered in this paper, the mortar method does achieve optimal convergence. Moreover, we presented several compulational strategies that do not require inversion of an interface matrix.

5.2 *Iteration on interface variables*

An alternative iterative strategy (Glowinski & Wheeler, 1988) uses the interface variables as the primary variables. If we combine the u and p variables into the symbol ψ, then system (2.7) can be written as

$$\begin{bmatrix} \mathscr{A}_L & 0 & \mathscr{C}_L \\ 0 & \mathscr{A}_R & \mathscr{C}_R \\ \mathscr{C}_L^T & \mathscr{C}_R^T & 0 \end{bmatrix} \begin{bmatrix} \psi_L \\ \psi_R \\ \xi \end{bmatrix} = \begin{bmatrix} \mathscr{F}_L \\ \mathscr{F}_R \\ 0 \end{bmatrix}. \quad (5.5)$$

We eliminate ψ as

$$\psi_i = \mathscr{A}_i^{-1}(\mathscr{F}_i - \mathscr{C}_i\xi), \quad i = L,R,$$

which gives the following system for ξ:

$$(\mathscr{C}_L^T\mathscr{A}_L^{-1}\mathscr{C}_L + \mathscr{C}_R^T\mathscr{A}_R^{-1}\mathscr{C}_R)\xi = (\mathscr{C}_L^T\mathscr{A}_L^{-1}\mathscr{F}_L + \mathscr{C}_R^T\mathscr{A}_R^{-1}\mathscr{F}_R). \quad (5.6)$$

If a Krylov method is applied to system (5.6), then only matrix vector products involving the matrix on the left are required. Since this matrix contains $\mathscr{A}_L^{-1}$ and $\mathscr{A}_R^{-1}$, obtaining a matrix vector product amounts to performing single domain component solves. Once ξ is obtained, ψ is recovered as above.

In the setting of geometric coupling, we rewrite the geometric finite volume system as

$$\begin{bmatrix} A_L & 0 & U_D & 0 \\ 0 & A_R & 0 & U_N \\ 0 & E_D & -I & 0 \\ E_N & 0 & 0 & -I \end{bmatrix} \begin{bmatrix} p_L \\ p_R \\ D \\ N \end{bmatrix} = \begin{bmatrix} F_L \\ F_R \\ 0 \\ 0 \end{bmatrix}, \tag{5.7}$$

where A_L and A_R are single domain finite volume systems, and the coupling strategy by which Dirichlet (D) and Neumann (N) data is provided by the opposite subdomain is defined by

$$E_N P_L = N \text{ and } E_D P_R = D.$$

Eliminating D and N from system (5.7) gives

$$\begin{bmatrix} A_L & U_D E_D \\ U_N E_N & A_R \end{bmatrix} \begin{bmatrix} p_L \\ p_R \end{bmatrix} = \begin{bmatrix} F_L \\ F_R \end{bmatrix},$$

which is identical to (2.10). If instead we eliminate p_L and p_R, the system (5.7) becomes

$$\begin{bmatrix} I & E_D A_R^{-1} U_N \\ E_N A_L^{-1} U_D & I \end{bmatrix} \begin{bmatrix} D \\ N \end{bmatrix} = \begin{bmatrix} E_D A_R^{-1} F_R \\ E_N A_L^{-1} F_L \end{bmatrix}, \tag{5.8}$$

which allows for an iteration of the form of (5.1) on the values D and N, from which the primary variables can be recovered. Solving (5.8) by iteration is analogous to solving (5.6) by iteration, and both require only component solves.

6. Conclusion

The geometric finite volume method (2.10) is often used to discretize problems on mismatched grids. Using the fact that the finite volume method is equivalent to the mixed finite element method with a certain quadrature, we have cast (2.10) in a form analogous to a mixed finite element mortar method. Doing so allows us to directly compare the two discretizations with an a-posteriori error estimate. We have shown with numerical examples that while the geometric finite volume method performs well in some cases, there are cases in which the performance deteriorates dramatically relative to the mixed finite element mortar method. Furthermore, such cases are not limited to problems in which the solution is changing rapidly near the interface. The deterioration was shown to be due mainly to incorrect transfer of information (or projection error) across the interface. Finally, we have shown that the mixed finite element mortar method can be paired with iterative solvers in such a way that it can be viewed as an alternative coupling strategy, requiring only single domain component solves at each iteration.

References

ARBOGAST, T., COWSAR, L. C., WHEELER, M. F. & YOTOV, I. (2000) Mixed finite element methods on nonmatching multiblock grids. *SIAM J. Numer. Anal.*, **37:4**, 1295–1315.

ARBOGAST, T., PENCHEVA, G., WHEELER, M. F. & YOTOV, I. (2007) A multiscale mortar mixed finite element method. *Multiscale Model. Simul.*, **6:1**, 319–346.

BECKER, R. & RANNACHER, R. (2001) An optimal control approach to a posteriori error estimation in finite element methods. *Acta Numer.*, **10**, 1–102.

BEN BELGACEM, F. (2000) The mixed mortar finite element method for the incompressible Stokes problem: convergence analysis. *SIAM J. Numer. Anal.*, **37**, 1085–1100.

BERNARDI, C., MADAY, Y. & PATERA, A. T. (1994) A new nonconforming approach to domain decomposition: The mortar element method. *Nonlinear partial differential equations and their applications*. UK: Longman Scientific and Technical.

BERNARDI, C., MADAY, Y. & RAPETTI, F. (2005) Basics and some applications of the mortar element method. *GAMM-Mitt.*, **28**, 97–123.

BREZZI, F. & FORTIN, M. (1991) *Mixed and hybrid finite element methods*. New York: Springer-Verlag.

CARY, J. R., CANDY, J., COHEN, R. H. *et al.* (2008) First results from core-edge parallel composition in the facets project. *J. Physics Conf. Series*, **125**. Fourth Annual Scientific Discovery Through Advanced Computing Conference (SciDAC 2008).

CARY, J. R. *et al.* (2010) Facets - a framework for parallel coupling of fusion components. *The 18th Euromicro International Conference on Parallel, Distributed and Network-Based Computing*. Pisa, Italy: IEEE, pp. 435–442.

EDWARDS, G. & ROGERS, C. (1998) Finite volume discretization with imposed flux continuity for the general tensor pressure equation. *Computational Geosciences*, **2**, 259–290.

ESTEP, D., LARSON, M. G. & WILLIAMS, R. D. (2000) Estimating the error of numerical solutions of systems of reaction-diffusion equations. *Mem. Amer. Math. Soc.*, **146**, viii+109.

ESTEP, D., TAVENER, S. & WILDEY, T. (2008) A posteriori analysis and improved accuracy for an operator decomposition solution of a conjugate heat transfer problem. *SIAM J. Numer. Analysis*, **46**, 2068–2089.

ESTEP, D., PERNICE, M., PHAM, D., TAVENER, S. & WANG, H. (2009a) A posteriori analysis of a cell-centered finite volume method for semilinear elliptic problems. *Journal of Computational and Applied Mathematics*, **233:2**.

ESTEP, D., TAVENER, S. & WILDEY, T. (2009b) A posteriori error analysis for a transient conjugate heat transfer problem. *Fin. El. Analysis Design*, **45**, 263–271.

ESTEP, D., TAVENER, S. & WILDEY, T. (2010) A posteriori error estimation and adaptive mesh refinement for a multi-discretization operator decomposition approach to fluid-solid heat transfer. *Journal of Computational Physics*, **229**, 4143–4158.

FARHAT, C., M., L. & LETALLEC, P. (1998) Load and motion transfer algorithms for fluid/ structure interaction problems with non-matching discrete interfaces: Momentum and energy conservation, optimal discretization and application to aeroelasticity. *Comput. Methods Appl. Mech. Engrg.*, **157**, 95–114.

GAIFFE, S., GLOWINSKI, R. & MASSON, R. (2002) Domain decomposition and splitting methods for mortar mixed finite element approximations to parabolic equations. *Numer. Math.*, **93:1**, 53–75.

GANIS, B. & YOTOV, I. (2009) Implementation of a mortar mixed finite element method using a multiscale flux basis. *Comp. Meth. in Appl. Mech. and Engng.*, **198:49**, 3989–3998.

GILES, M. B. & SULI, E. (2002) Adjoint methods for pdes: a posteriori error analysis and postprocessing by duality. *Acta Numer.*, **11**, 145–236.

GLOWINSKI, R. & WHEELER, M. F. (1988) Domain decomposition and mixed finite element methods for elliptic problems. *First International Symposium on Domain Decomposition Methods for Partial Differential Equations* (R. Glowinski, G. H. Golub, G. A. Meurant & J. Periaux eds). Philadelphia: SIAM, pp. 144–172.

HANSBRO, P. & LARSON, M. G. (2011) A posteriori error estimates for continuous/discontinuous galerkin approximations of the Kirchhoff-Love plate. *Comp. Meth. in Appl. Mech. and Engng.*, **200:47-48**, 3289–3295.

QUARTERONI, A., PASQUARELLI, F. & VALLI, A. (1992) Heterogeneous domain decomposition: principles, algorithms, applications. *Fifth International Symposium on Domain Decomposition Methods for Partial Differential Equations*. Philadelphia: SIAM, pp. 129–150.

ROBERTS, J. E. & THOMAS, J. (1991) Mixed and hybrid methods. *Handbook of numerical analysis*. Amsterdam: Elsevier Science Publishers B.V., pp. 523–639.

RUSSELL, T. & WHEELER, M. F. (1983) Finite element and finite difference methods for continuous flows in porous media. *The Mathematics of Reservoir Simulation* (R. E. Ewing ed.). Philadelphia: SIAM, pp. 35–106.

WEISER, A. & WHEELER, M. F. (1988) On convergence of block-centered finite differences for elliptic problems. *SIAM J. Numer. Anal.*, **25**, 351–375.

WHEELER, M. F. & YOTOV, I. (2005) A posteriori error estimates for the mortar mixed finite element method. *SIAM J. Numer. Anal.*, **43**, 1021–1042.

A posteriori analysis of an iterative multi-discretization method for reaction-diffusion systems

J. H. Chaudhry[1], D. Estep[2], V. Ginting[3], S. Tavener[4]

[1] *Department of Mathematics, Colorado State University, Fort Collins, CO 80523.*
J. H. Chaudhry's work is supported in part by the Department of Energy (DE-SC0005304).
[2] Corresponding Author, *Department of Statistics, Colorado State University, Fort Collins, CO 80523; estep@stat.colostate.edu; (phone) 970-491-6722; (fax) 970-491-7895*
D. Estep's work is supported in part by the Defense Threat Reduction Agency (HDTRA1-09-1-0036), Department of Energy (DE-FG02-04ER25620, DE-FG02-05ER25699, DE-FC02-07ER54909, DE-SC0001724, DE-SC0005304, INL00120133, DE0000000SC9279), Idaho National Laboratory (00069249, 00115474), Lawrence Livermore National Laboratory (B573139, B584647, B590495), National Science Foundation (DMS-0107832, DMS-0715135, DGE-0221595003, MSPA-CSE-0434354, ECCS-0700559, DMS-1065046, DMS-1016268, DMS-FRG-1065046, DMS-1228206), National Institutes of Health (#R01GM096192).
[3] *Department of Mathematics, University of Wyoming, Laramie, WY 82071.*
V. Ginting's work is supported in part by the National Science Foundation (DMS-1016283), the Department of Energy (DE-SC0004982)
[4] *Department of Mathematics, Colorado State University, Fort Collins, CO 80523*

Abstract

This paper is concerned with the accurate computational error estimation of numerical solutions of multi-scale, multi-physics systems of reaction-diffusion equations. Such systems can present significantly different temporal and spatial scales within the components of the model, indicating the use of independent discretizations for different components. However, multi-discretization can have significant effects on accuracy and stability. We perform an adjoint-based analysis to derive asymptotically accurate *a posteriori* error estimates for a user-defined quantity of interest. These estimates account for leading order contributions to the error arising from numerical solution of each component, an error due to incomplete iteration, an error due to linearization, and for errors arising due to the projection of solution components between different spatial meshes. Several numerical examples with various settings are given to demonstrate the performance of the error estimators.

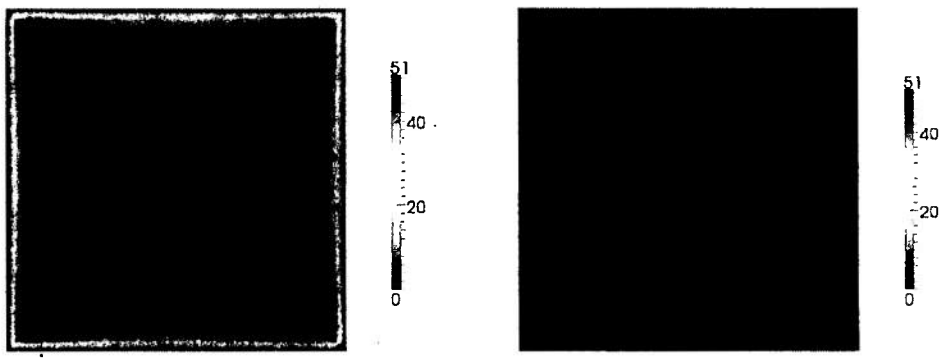
Keywords: reaction-diffusion, adjoint operator, *a posteriori* estimates, discontinuous Galerkin method, iterative method, multirate method, multi-scale discretization, operator decomposition

1. Introduction

This paper is concerned with the accurate computational error estimation of numerical solutions of multi-scale, multi-physics systems of reaction-diffusion equations. The components of solutions of such multi-scale, multi-physics models typically exhibit spatial and temporal behavior occurring over a significant range of scales. For example, consider the well-known Brusselator model for chemical dynamics [25, 1]. This is a system of reaction-diffusion equations whose separate components can behave over different spatial and temporal scales for particular choices of parameters. The model is

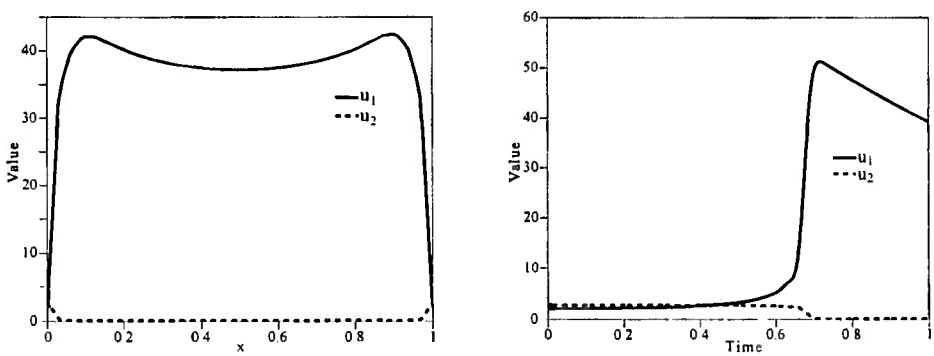
$$\begin{aligned} \dot{u}_1 - \epsilon_1 \Delta u_1 &= \alpha - (\beta+1)u_1 + u_1^2 u_2, && x \in \Omega \subset \mathbb{R}^2, t > 0, \\ c\dot{u}_2 - \epsilon_2 \Delta u_2 &= \beta u_1 - u_1^2 u_2, && x \in \Omega, t > 0, \\ u_1(x,t) &= \alpha,\ u_2(x,t) = \beta/\alpha, && x \in \partial\Omega, t > 0, \\ u_1(x,0) &= u_{1,0}(x),\ u_2(x,0) = u_{2,0}(x), && x \in \Omega, \end{aligned} \tag{1}$$

where u_1 and u_2 are concentrations of species 1 and 2, respectively. We assume that $0 < \epsilon_0 < \epsilon_i$, $i = 1,2$, for some positive constant ϵ_0. The solutions are multi-scale in time and space for a wide range of parameter values. In Fig. 1 we show the solution at $t = 1.0$ corresponding to $\alpha = 2$, $\beta = 5.45$, $\epsilon_1 = 0.008$, $\epsilon_2 = 0.08$, $c = 20$ and initial conditions $u_{1,0}(x) = \alpha + 0.1\sin(\pi x_1)\sin(\pi x_2)$ and $u_{2,0}(x) = \beta/\alpha + 0.1\sin(\pi x_1)\sin(\pi x_2)$. We plot a cross section of the numerical solution at $x_2 = 0.25$ in Fig. 2. There are sharp spatial gradients for the component u_1, while u_2 shows relatively less spatial variation, suggesting that we might use an relatively finer mesh to resolve u_1. The time evolution of the solution at the point $x = (0.25, 0.25)$ is also shown in Fig. 2 and indicates the multirate nature of the solutions in which u_1 is a faster component than u_2 and requires relatively fine time steps for accurate resolution.



(a) (b)

Figure 1: Brusselator: Color contour plots of the solution at $T = 1.0$. (a) $u_1(x)$. (b) $u_2(x)$.



(a) (b)

Figure 2: Brusselator. (a) Spatial cross section of the solution at $x_2 = 0.25$ and $T = 1.0$. (b) Temporal cross section of the solution at $x = (0.25, 0.25)$.

In practical situations, the error of approximate solutions of multi-scale, multi-physics evolution models is always significant. Simply providing an *a priori* analysis of convergence and an assertion that the error is small for sufficiently refined discretizations that cannot be achieved in practice is inadequate for scientific purposes. Hence, application of numerical solution to predictive science and engineering applications requires accurate estimation of information computed from numerical solution as part of the overall uncertainty quantification critical to scientific and engineering needs.

For multi-scale problems, the demands of computational efficiency (or simple necessity) suggests a multi-discretization approach that involves solving the distinct components of a multi-physics model using independent meshes and time steps chosen to resolve behavior on the pertinent scales. A multi-

discretization strategy often has significant effects on the accuracy and stability of the numerical solution. Indeed, such multi-discretization methods fall into the general class of multi-scale operator decomposition methods [11], that typically employ some form of projection to link solutions computed on different spatial and temporal meshes and necessarily "synchronize" solutions that have been decoupled during an iterative solution process. Since these practices can have a complex effect on accuracy and stability, there has been a steady development of *a posteriori* error estimates for a wide range of multi-scale operator decomposition methods in recent years [13, 12, 16, 18, 6, 7, 22, 21, 23] extending earlier work on *a posteriori* error analysis employing computable residuals and adjoint problems, see e.g. [10, 8, 9, 15, 19, 5, 3, 4]. While the primary purpose of such estimates is to quantify the contributions of various sources of discretization error on accuracy and stability, the estimates can also provide guidance as to the choice of numerical parameters needed to obtain a desired accuracy.

The analysis of multi-discretization numerical methods for multi-scale systems of partial differential equations in this paper extends earlier results for multi-rate time integration schemes for initial value problems for ordinary differential equations in [14]. For simplicity, we consider a system comprised of two reaction-diffusion equations: Find $u = (u_1\ u_2)^\top$ that satisfies

$$\begin{aligned} \dot{u}_1 - \nabla\cdot(\epsilon_1\nabla u_1) &= f_1(u_1,u_2), && (\boldsymbol{x},t)\in\Omega\times(0,T],\\ \dot{u}_2 - \nabla\cdot(\epsilon_2\nabla u_2) &= f_2(u_1,u_2), && (\boldsymbol{x},t)\in\Omega\times(0,T],\\ u_i(\boldsymbol{x},t) &= 0, && (\boldsymbol{x},t)\in\partial\Omega\times(0,T],\ i=1,2,\\ u_i(\boldsymbol{x},0) &= g_i(\boldsymbol{x}), && \boldsymbol{x}\in\Omega,\ i=1,2, \end{aligned} \tag{2}$$

where Ω is a convex polygonal domain with boundary $\partial\Omega$, $\{f_i\}$ are differentiable functions of their arguments, $\{\epsilon_i\}$ and $\{g_i\}$ are smooth functions in Ω, and there is a constant $\epsilon_0 > 0$ such that $\epsilon_i \geq \epsilon_0 > 0$ on Ω. Finally, we also assume that

$$f_i(0) = 0, \quad i = 1,2. \tag{3}$$

The latter assumption is used to define the adjoint problems employed for the *a posteriori* error analysis carried out in Sec. 4. The ideas and results extend to systems consisting of more than two equations in a straightforward way. Condition (3) can also be generalized, see Sec. 4. Finally, neglecting the vastly more difficult questions of existence, uniqueness, and regularity for the problem, the analysis also extends to problems with nonlinear diffusion constants, and we show the formal result in Sec. 7.

Whenever appropriate, we write the differential equations in a compact form

$$\dot{u} - \nabla\cdot(\epsilon\nabla u) = f(u),$$

where $\epsilon = \mathrm{diag}(\epsilon_1,\epsilon_2)$, $\nabla u = [\nabla u_1\ \ \nabla u_2]^\top$, and $f(u) = [f_1(u)\ \ f_2(u)]^\top$. The diffusion coefficients, ϵ_1 and ϵ_2, and reaction terms f_1 and f_2 may induce different spatial and temporal properties for u_1 and u_2. We adopt a multi-discretization approach in which each component model is solved on its own scale. In order to facilitate this approach, we compute the solution using a common iterative approach in which each component model is solved while fixing the other component solutions. The individual component solves are synchronized by exchanging information at designated "synchronization" times. At each synchronization time, component exchanges are iterated a specified number of times before the solution proceeds to the next synchronization time.

In this paper, we derive accurate *a posteriori* error estimates for a quantity of interest obtained from a numerical solution computed using the iterative multi-discretization scheme. The estimates account for *leading order* contributions to the error arising from numerical solution of each component, multi-discretization, and iterative solution. The estimates quantify the relative size of the various contributions to the error. We demonstrate the accuracy of the estimates on a variety of examples.

The rest of the paper is organized as follows. In Sec. 2, we formulate an iterative multi-discretization Galerkin finite element method for (2). In Sec. 3, we formulate an *analytic* version of (2) that we use for

the purpose of analysis. We present the first results of an analysis for the multi-discretization solution method in Sec. 4 followed by numerical examples in Sec. 5. In Sec. 6, we expand the analysis to include the effects of using different space meshes for the two components. We also give numerical results for the Brusselator problem in this section. Finally, in Sec. 7 we consider the analysis for systems in which the diffusion coefficient may depend on the solution.

2. An iterative multi-discretization Galerkin finite element method

In Alg. 1 we formulate the iterative multi-discretization Galerkin finite element method for (2). We first discretize $[0,T]$ into $0 = t_0 < t_1 < t_2 < \cdots < t_N = T$ with time steps $\{\Delta t_n = t_n - t_{n-1}\}_{n=1}^N$, $\Delta t = \max_{1\le n\le N}\{\Delta t_n\}$ and time intervals $I_n = [t_{n-1}, t_n]$. We think of $\{t_n\}$ as synchronization times during which information between the two component solves interior to the nodes is exchanged iteratively. To each t_n, we assign a positive integer M_n which is the number of iterations to be used when synchronizing the fast and slow components.

To solve the components over each synchronization interval, we divide the intervals $\{I_n\}$ into a number of smaller time steps. We let $L_{i,n}$, $i = 1,2$ be two positive integers, where $L_{1,n}$ denotes the number of time steps used to solve the subsystem 1 and $L_{2,n}$ the number of steps used for subsystem 2 on each synchronization interval. Without loss of generality, we assume $L_{1,n} = d_n L_{2,n}$ for some positive integer d_n, i.e., $L_{1,n}$ is divisible by $L_{2,n}$. We denote time steps for each component in the Galerkin formulation by $\Delta s_{i,n} = \Delta t_n / L_{i,n}$, with $\Delta s_i = \max_{1\le n\le N}\{\Delta s_{i,n}\}$. We use an extension of the discontinuous Galerkin method [15]. The method naturally extends to the continuous Galerkin method [15].

To construct the finite dimensional spaces, we first discretize Ω into triangulations $\mathcal{T}_{h_i}$, where h_i denotes the maximum diameter of the elements of $\mathcal{T}_{h_i}$, $i = 1,2$, i.e., each equation has different triangulation. Each of these triangulations is arranged in such a way that the union of the elements of $\mathcal{T}_{h_i}$ is Ω, and the intersection of any two elements is either a common edge, node, or is empty.

The approximations are polynomials in time and continuous piecewise polynomials in space on each space-time slab $S_{l,n} = \Omega \times I_{l,n}$, for $l = 1,\cdots,L_{1,n}$ and $S_{k,n} = \Omega \times I_{k,n}$, for $k = 1,\cdots,L_{2,n}$. Here $I_{l,n} = [t_{n-1}+(l-1)\Delta s_{1,n}, t_{n-1}+l\Delta s_{1,n}]$ and $I_{k,n} = [t_{n-1}+(k-1)\Delta s_{2,n}, t_{n-1}+k\Delta s_{2,n}]$ are the smaller time intervals. In space, we let $V_{h_i} \subset H_0^1(\Omega)$ denote the space of continuous piecewise polynomial functions $v(x) \in \mathbb{R}$ defined on $\mathcal{T}_{h_i}$. (For simplicity we confine our attention to problems with homogeneous Dirichlet boundary conditions). On each slab, we define

$$W_{l,n}^{q_1} = \left\{ w(x,t) : w(x,t) = \sum_{j=0}^{q_1} t^j v_j(x), \quad v_j \in V_{h_1}, \quad (x,t) \in S_{l,n} \right\},$$

$$W_{k,n}^{q_2} = \left\{ w(x,t) : w(x,t) = \sum_{j=0}^{q_2} t^j v_j(x), \quad v_j \in V_{h_2}, \quad (x,t) \in S_{k,n} \right\}.$$

We denote the jump across t_n by $[w]_n = w_n^+ - w_n^-$, where $w_n^\pm = \lim_{s\to t_n^\pm} w(s)$. We let $\Pi_{1\to 2} : W_{l,n}^{q_1} \to W_{k,n}^{q_2}$, $\Pi_{2\to 1} : W_{k,n}^{q_2} \to W_{l,n}^{q_1}$ denote projections between the two spaces. The iterative discontinuous Galerkin dG(q) finite element approximation is written down in Alg. 1. In the algorithm , $U^{(m)} = [U_1^{(m)}, U_2^{(m)}]^\top \in W_{l,n}^{q_1} \times W_{k,n}^{q_2}$ are the finite element solutions, defined locally on time intervals $I_{l,n}$ and $I_{k,n}$. The notation (a,b) denotes the L^2 inner product, or simply the spatial integral, $\int_\Omega a\,b\,dx$.

3. An analytic iterative method

The approach to the *a posteriori* analysis of the multi-discretization finite element approximation in Alg. 1 we use in this paper starts with defining an iterative method to determine an *analytic* solution

Algorithm 1 Iterative multi-discretization Galerkin finite element method

Set $U^{(M_0)}(\cdot, t_0^-) = u(\cdot, t_0)$
for $n = 1$ to N **do**
  Set $U_2^{(0)} = U_2^{(M_{n-1})}(\cdot, t_{n-1})$
  for $m = 1$ to M_n **do**
    Set $U^{(m)}(\cdot, t_{n-1}^-) = U^{(M_{n-1})}(\cdot, t_{n-1}^-)$
    for $l = 1$ to $L_{1,n}$ **do**
      Compute $U_1^{(m)} \in W_{l,n}^{q_1}$ satisfying

$$\int_{I_{l,n}} \left(\dot{U}_1^{(m)} - f_1(U_1^{(m)}, \Pi_{2\to 1} U_2^{(m-1)}), V \right) dt + \int_{I_{l,n}} \left(\epsilon_1 \nabla U_1^{(m)}, \nabla V \right) dt + \left([U_1^{(m)}]_{l-1,n}, V_{l-1}^+ \right) = 0 \qquad (4)$$

      for all $V \in W_{l,n}^{q_1}$
    end for
    for $k = 1$ to $L_{2,n}$ **do**
      Compute $U_2^{(m)} \in W_{k,n}^{q_2}$ satisfying

$$\int_{I_{k,n}} \left(\dot{U}_2^{(m)} - f_2(\Pi_{1\to 2} U_1^{(m)}, U_2^{(m)}), Z \right) dt + \int_{I_{k,n}} \left(\epsilon_2 \nabla U_2^{(m)}, \nabla Z \right) dt + \left([U_2^{(m)}]_{k-1,n}, Z_{k-1}^+ \right) = 0 \qquad (5)$$

      for all $Z \in W_{k,n}^{q_2}$
    end for
  end for
end for

of (2) obtained via a sequence of functions $\{u_i^{(m)}(t)\}$ that map the time intervals to the Banach space $X = L^2(\Omega)$, i.e., $u_i^{(m)}(t) : [t_{n-1}, t_n] \times X \to X$ for $i = 1,2$. The iterative method defining $\{u_i^{(m)}\}$ is given in Alg. 2.

The accuracy of the computational error estimate derived below assumes that the analytic iteration has converged to a sufficient extent and the discretization error is sufficiently small. The following assumptions provide sufficient general conditions to guarantee convergence of $u_i^{(m)}$ to u_i, $i = 1,2$:

Assumption A.1. *Assume that $f(t,u) : [t_{n-1}, t_n] \times X \times X \to X \times X$ is uniformly Lipschitz continuous with constant L, i.e.*

$$\|f(t,u) - f(t,v)\|_{X\times X} \le L\|u - v\|_{X\times X} \quad \forall t \ge 0. \qquad (8)$$

Similarly, we assume that $f'(u)$ is uniformly Lipschitz continuous with constant L'.'

Assumption A.2. *Let M be the bound on the semigroup G associated with* (2) *(defined in the Appendix). We assume that the time steps Δt_n satisfy the inequality,*

$$ML\Delta t_n \exp(ML\Delta t_n) < 1 \qquad (9)$$

The convergence proof is given in the Appendix. We note that these are sufficient conditions to guarantee convergence of the iteration. They are not necessary and the iteration may converge in specific cases without satisfying these assumptions. Our *a posteriori* analysis assumes the iteration is convergent and employs the Lipschitz assumptions, but does not specifically depend on the bound on the semigroup.

Algorithm 2 Analytic iterative method

for $n = 1$ to N **do**
  Set $u_2^{(0)} = u_2^{(M_{n-1})}(\cdot, t_{n-1})$
  for $m = 1$ to M_n **do**
    Compute $u_1^{(m)}(\boldsymbol{x}, t)$ in $\Omega \times I_n$ satisfying

$$\begin{cases} \dot{u}_1^{(m)} - \nabla \cdot \left(\epsilon_1 \nabla u_1^{(m)}\right) = f_1\left(u_1^{(m)}, u_2^{(m-1)}\right), & (\boldsymbol{x}, t) \in \Omega \times I_n, \\ u_1^{(m)}(\boldsymbol{x}, t) = 0, & (\boldsymbol{x}, t) \in \partial\Omega \times I_n, \\ u_1^{(m)}(\boldsymbol{x}, t_{n-1}) = u_1^{(M_{n-1})}(\boldsymbol{x}, t_{n-1}), & \boldsymbol{x} \in \Omega. \end{cases} \tag{6}$$

    Compute $u_2^{(m)}(\boldsymbol{x}, t)$ in $\Omega \times I_n$ satisfying

$$\begin{cases} \dot{u}_2^{(m)} - \nabla \cdot \left(\epsilon_2 \nabla u_2^{(m)}\right) = f_2\left(u_1^{(m)}, u_2^{(m)}\right), & (\boldsymbol{x}, t) \in \Omega \times I_n, \\ u_2^{(m)}(\boldsymbol{x}, t) = 0, & (\boldsymbol{x}, t) \in \partial\Omega \times I_n, \\ u_2^{(m)}(\boldsymbol{x}, t_{n-1}) = u_2^{(M_{n-1})}(\boldsymbol{x}, t_{n-1}), & \boldsymbol{x} \in \Omega. \end{cases} \tag{7}$$

  end for
end for

The motivation for introducing the analytic iterative solution method is the realization that the iterative multi-discretization Galerkin finite element method in Alg. 1 is a consistent finite element space-time discretization of Alg. 2. In particular, in (4) and (5) we have chosen piecewise space-time polynomials that solve the weak or variational formulation of (6) and (7) respectively. The variational formulation is obtained by multiplying each (6) and (7) by appropriate test functions, integrating over space and time, and using Green's formula on the elliptic part. In practice, we evaluate the finite element function using quadrature to approximate the associated integral, which results in a set of discrete equations.

4. *A posteriori* analysis of the iterative multi-discretization Galerkin finite element method

We derive computational *a posteriori* error estimates based on variational analysis, residuals of the finite element approximation, and the generalized Green's function solving the adjoint problem [8, 10, 9, 15, 19, 5, 3, 11, 4]. We first develop the analysis assuming the same spatial meshes for both components. We relax this restriction in Sec. 6 where we include the effect of projection between different spatial meshes.

A key feature of the analysis is the realization that the iterative multi-discretization approximation is naturally associated with a **different** adjoint operator than that for the original problem. For this reason, we use a different linearization than commonly employed for nonlinear problems [13]. We assume that the operators for the original problem and the analytic operator decomposition version share a common solution, and use that as a linearization point for determining the stability properties of solutions in the neighborhood of the linearization point. The simplest example is to assume a common steady-state solution such as 0, which is guaranteed by the homogeneity assumption (3), i.e., $f(0) = 0$. This assumption is employed in many standard analyses of the model (2) and it is satisfied in a great many cases. The condition can be generalized ([see 13]), e.g. to other steady state solutions or to a given function of time. We give an example of a system (Brusselator) that uses an alternative condition in Sec. 6 [13]. We let

$$\overline{f'_{ij}(u)} = \int_0^1 \frac{\partial f_i}{\partial u_j}(su)\, ds, \quad i, j = 1, 2, \tag{10}$$

and $\overline{f'(u)}$ denotes the square matrix whose entries are (10). Then $f(u) = \overline{f'(u)}u$. Associated with this linearized form, we denote by φ, the generalized Green's function satisfying the following adjoint problem:

$$\begin{cases} -\dot{\varphi} - \nabla\cdot(\epsilon\nabla\varphi) = \overline{f'(u)}^{\top}\varphi, & (\boldsymbol{x},t)\in\Omega\times(T,0], \\ \varphi(\boldsymbol{x},t) = 0, & (\boldsymbol{x},t)\in\partial\Omega\times(T,0], \\ \varphi(\boldsymbol{x},T) = \psi(\boldsymbol{x}), & \boldsymbol{x}\in\Omega, \end{cases} \qquad \epsilon\nabla\varphi = \begin{pmatrix}\epsilon_1\nabla\varphi_1\\ \epsilon_2\nabla\varphi_2\end{pmatrix}. \tag{11}$$

On subinterval $I_n = (t_{n-1}, t_n)$, we define the solution operators Φ_n associated with the Green's function,

$$\varphi(x,t) = \Phi_n(t)\psi_n(x),$$

for $t_n > t \geq t_{n-1}$ and some initial data ψ_n. To get solution representation using the Green's functions, we multiply u with (11), integrate in time and space, resulting in

$$\begin{aligned}(u_n,\psi_n) &= (u_{n-1},\varphi_{n-1}) + \int_{I_n}\left(\dot{u} - \nabla\cdot(\epsilon\nabla u) - \overline{f'(u)}u,\varphi\right)dt \\ &= (u_{n-1},\varphi_{n-1}) + \int_{I_n}\left(\dot{u} - \nabla\cdot(\epsilon\nabla u) - f(u),\varphi\right)dt.\end{aligned} \tag{12}$$

Because u solves (2), this last equality gives

$$(u_n,\psi_n) = (u_{n-1},\Phi_n\psi_n). \tag{13}$$

4.1. Analysis of the analytic iterative method

To simplify presentation, we express the analytic iterative method in Alg. 2 in a more compact format. In particular, for any iteration index m, we write (6) and (7) as

$$\dot{u}^{(m)} - \nabla\cdot(\epsilon\nabla u^{(m)}) = f(u^{(m)}) + \delta_R^{(m)}, \qquad \delta_R^{(m)} = \begin{pmatrix} -\left(f_1(u_1^{(m)},u_2^{(m)}) - f_1(u_1^{(m)},u_2^{(m-1)})\right)\\ 0\end{pmatrix} \tag{14}$$

The vector $\delta_R^{(m)}$ can be interpreted as residuals at the iteration level m.

To define an adjoint for the approximation in Alg. 2, we let φ_i denote the generalized Green's function that satisfies an adjoint problem on time interval I_n as given in Alg. 3. Here K_n refers to the number of iterations to be used when synchronizing the two components of the adjoint.

Notice that the adjoint problems are solved backward in time and in the reverse order to that of the forward problem, starting with φ_2 followed by φ_1. These generalized Green's functions are an iterative approximation of (11). We note that the coefficients $\overline{f'_{ij}(u^{(m)})}$ are linearized around $u^{(m)}$. As in the forward problem, we can also rewrite this last algorithm into a compact form

$$-\dot{\varphi}^{(k)} - \nabla\cdot(\epsilon\nabla\varphi^{(k)}) = \overline{f'(u^{(m)})}^{\top}\varphi^{(k)} + \xi_R^{(k)}, \qquad \xi_R^{(k)} = \begin{pmatrix} 0 \\ -\left(\overline{f'_{12}(u^{(m)})}(\varphi_1^{(k)} - \varphi_1^{(k-1)})\right)\end{pmatrix}, \tag{17}$$

for adjoint iteration level k. Here, $\xi_R^{(k)}$ is the residual of the adjoint at iteration level k. We also introduce the solution operators $\Phi_n^{(k)}$, with $\varphi^{(k)}(\boldsymbol{x},t) = \Phi_n^{(k)}(t)\psi_n(\boldsymbol{x})$, for $t_n > t \geq t_{n-1}$. To get a representation of the iterative solution, we follow a similar derivation for the fully coupled problem (see (12)). Multiplying equation (17) by $u^{(m)}$, integrating each over $\Omega\times I_n$ and applying integration by parts in time, and Green's Theorem in space and using (14), we obtain the solution representation of the analytic iterative method

$$(u_n^{(m)},\psi_n) = (\dot{u}_{n-1}^{(m)},\Phi_n^{(k)}\psi_n) + \int_{I_n}\left(\delta_R^{(m)},\varphi^{(k)}\right)dt - \int_{I_n}\left(u^{(m)},\xi_R^{(k)}\right)dt. \tag{18}$$

Algorithm 3 Adjoint for the analytic iterative method

Set $\varphi_1^{(0)} = \psi_{1,n}$
for $k = 1$ to K_n **do**
Compute $\varphi_2^{(k)}(x,t)$ in $\Omega \times (t_n, t_{n-1}]$, satisfying

$$\begin{cases} -\dot{\varphi}_2^{(k)} - \nabla\cdot\left(\epsilon_2\cdot\nabla\varphi_2^{(k)}\right) = \overline{f'_{22}(u^{(m)})}\varphi_2^{(k)} + \overline{f'_{12}(u^{(m)})}\varphi_1^{(k-1)}, & (x,t)\in\Omega\times(t_n,t_{n-1}], \\ \varphi_2^{(k)}(x,t) = 0, & (x,t)\in\partial\Omega\times(t_n,t_{n-1}], \\ \varphi_2^{(k)}(x,t_n) = \psi_{2,n}(x), & x\in\Omega. \end{cases} \tag{15}$$

Compute $\varphi_1^{(k)}(x,t)$ in $\Omega \times (t_n, t_{n-1}]$, satisfying

$$\begin{cases} -\dot{\varphi}_1^{(k)} - \nabla\cdot\left(\epsilon_1\nabla\varphi_1^{(k)}\right) = \overline{f'_{11}(u^{(m)})}\varphi_1^{(k)} + \overline{f'_{21}(u^{(m)})}\varphi_2^{(k)}, & (x,t)\in\Omega\times(t_n,t_{n-1}], \\ \varphi_1^{(k)}(x,t) = 0, & (x,t)\in\partial\Omega\times(t_n,t_{n-1}], \\ \varphi_1^{(k)}(x,t_n) = \psi_{1,n}(x), & x\in\Omega. \end{cases} \tag{16}$$

end for

We note that this representation is not in the standard format (in which the solution at the current time level solely depends on the previous time level values). It contains remnants arising from the iterative procedure used to compute both forward and backward problems. The second term can be interpreted as the weighted average of the forward problem residual over a time step. The third term, on the other hand, is the weighted average of the backward problem residual over a time step. Thus, the iterative nature of solution procedure is reflected in this representation. Once convergence is reached both on forward and backward problems, then the standard convention of solution representation using the adjoint technique is recovered.

We are now able to express the error representation of the iterative implicit method. Let $\hat{e}_n^{(m)} = u_n - u_n^{(m)}$. Now, we state a lemma concerning an error equation over one time step.

Lemma 4.1. *The analytic iterative method satisfies the following error equation over one time step:*

$$(\hat{e}_n^{(m)},\psi_n) = (u_n - u_n^{(m)},\psi_n) = (\hat{e}_{n-1}^{(m)},\Phi_n^{(k)}\psi_n) + (\hat{e}_{n-1}^{(m)},\Delta\Phi_n\psi_n) + (u_{n-1}^{(m)},\Delta\Phi_n\psi_n) - \int_{I_n}\left(\delta_R^{(m)},\varphi^{(k)}\right)dt + \int_{I_n}\left(u^{(m)},\xi_R^{(k)}\right)dt,$$

where $\Delta\Phi_n = (\Phi_n - \Phi_n^{(k)})$.

Proof. Subtracting (18) from (13), and adding and subtracting $(u_{n-1}^{(m)},\Phi_n\psi_n)$,

$$\begin{aligned}(\hat{e}_n^{(m)},\psi_n) &= (u_n - u_n^{(m)},\psi_n) \\ &= (u_{n-1},\Phi_n\psi_n) - (u_{n-1}^{(m)},\Phi_n\psi_n) + (u_{n-1}^{(m)},\Phi_n\psi_n) - (u_{n-1}^{(m)},\Phi_n^{(k)}\psi_n) \\ &\quad - \int_{I_n}\left(\delta_R^{(m)},\varphi^{(k)}\right)dt + \int_{I_n}\left(u^{(m)},\xi_R^{(k)}\right)dt \\ &= (u_{n-1} - u_{n-1}^{(m)},\Phi_n\psi_n) + (u_{n-1}^{(m)},\Delta\Phi_n\psi_n) - \int_{I_n}\left(\delta_R^{(m)},\varphi^{(k)}\right)dt + \int_{I_n}\left(u^{(m)},\xi_R^{(k)}\right)dt.\end{aligned}$$

Adding and subtracting $(\hat{e}_{n-1}^{(m)},\Phi_n^{(k)}\psi_n)$ to above equation completes the proof. □

4.2. Analysis of the iterative multi-discretization Galerkin finite element method

To construct the adjoint, let $z^{(m)} = su^{(m)} + (1-s)U^{(m)}$, with $s \in [0,1]$. Then let $\overline{f'(z^{(m)})}$ be a matrix whose entries are

$$\overline{f'(z^{(m)})}_{ij} = \int_0^1 \frac{\partial f_i}{\partial u_j}(z^{(m)})\, ds.$$

Consequently, $f(u^{(m)}) - f(U^{(m)}) = \overline{f'(z^{(m)})}\,(u^{(m)} - U^{(m)})$.

Associated with the finite element solution, we denote by ϑ the generalized Green's function that satisfies the adjoint problem in Alg. 4. As was the case in the adjoint formulation associated with ana-

Algorithm 4 Adjoint for the iterative multi-discretization Galerkin finite element method

Set $\vartheta_1^{(0)} = \psi_{1,n}$
for $k = 1$ to K_n **do**
Compute $\vartheta_2^{(k)}(\boldsymbol{x}, t)$ in $\Omega \times (t_n, t_{n-1}]$ satisfying

$$\begin{cases} -\dot{\vartheta}_2^{(k)} - \nabla\cdot\left(\epsilon_2 \nabla \vartheta_2^{(k)}\right) = \overline{f'_{22}(z^{(m)})}\vartheta_2^{(k)} + \overline{f'_{12}(z^{(m)})}\vartheta_1^{(k-1)}, & (\boldsymbol{x}, t) \in \Omega \times (t_n, t_{n-1}], \\ \vartheta_2^{(k)}(\boldsymbol{x}, t) = 0, & (\boldsymbol{x}, t) \in \partial\Omega \times (t_n, t_{n-1}], \\ \vartheta_2^{(k)}(\boldsymbol{x}, t_n) = \psi_{2,n}(\boldsymbol{x}), & \boldsymbol{x} \in \Omega. \end{cases} \tag{19}$$

Compute $\vartheta_1^{(k)}(\boldsymbol{x}, t)$ in $\Omega \times (t_n, t_{n-1}]$ satisfying

$$\begin{cases} -\dot{\vartheta}_1^{(k)} - \nabla\cdot\left(\epsilon_1 \nabla \vartheta_1^{(k)}\right) = \overline{f'_{11}(z^{(m)})}\vartheta_1^{(k)} + \overline{f'_{21}(z^{(m)})}\vartheta_2^{(k)}, & (\boldsymbol{x}, t) \in \Omega \times (t_n, t_{n-1}], \\ \vartheta_1^{(k)}(\boldsymbol{x}, t) = 0, & (\boldsymbol{x}, t) \in \partial\Omega \times (t_n, t_{n-1}], \\ \vartheta_1^{(k)}(\boldsymbol{x}, t_n) = \psi_{1,n}(\boldsymbol{x}), & \boldsymbol{x} \in \Omega. \end{cases} \tag{20}$$

end for

lytic iterative method, this algorithm can be expressed as a compact form

$$-\dot{\vartheta}^{(k)} - \nabla\cdot(\epsilon\nabla\vartheta^{(k)}) = \overline{f'(z^{(m)})}^{\mathsf{T}}\vartheta^{(k)} + \eta_R^{(k)}, \qquad \eta_R^{(k)} = \begin{pmatrix} 0 \\ -\left(\overline{f'_{12}(z^{(m)})}(\vartheta_1^{(k)} - \vartheta_1^{(k-1)})\right). \end{pmatrix} \tag{21}$$

Here, $\eta_R^{(k)}$ is the residual of the adjoint at iteration level k.

At this stage, we are in position to derive an error equation associated with the iterative multi-discretization Galerkin finite element method. Let $\tilde{e}^{(m)} = u^{(m)} - U^{(m)}$. First notice that using integration by parts,

$$\left(\epsilon\nabla u^{(m)} - \epsilon\nabla U^{(m)}, \nabla\vartheta^{(k)}\right) = \left(\epsilon\nabla\tilde{e}^{(m)}, \nabla\vartheta^{(k)}\right) \quad = \left(\tilde{e}^{(m)}, -\nabla\cdot(\epsilon\nabla\vartheta^{(k)})\right).$$

Similarly,

$$\left(f(u^{(m)}) - f(U^{(m)}), \vartheta^{(k)}\right) = \left(\overline{f'(z^{(m)})}\,\tilde{e}^{(m)}, \vartheta^{(k)}\right) = \left(\tilde{e}^{(m)}, \overline{f'(z^{(m)})}^{\mathsf{T}}\vartheta^{(k)}\right).$$

Furthermore, using continuity of $u^{(m)}$,

$$\tilde{e}_{l-1,n}^{(m)+} = u_{l-1,n}^{(m)+} - U_{l-1,n}^{(m)+} = (u_{l-1,n}^{(m)-} - U_{l-1,n}^{(m)-}) - (U_{l-1,n}^{(m)+} - U_{l-1,n}^{(m)-}) = \tilde{e}_{l-1,n}^{(m)-} - [U^{(m)}]_{l-1,n}.$$

We use these three expressions on time interval $I_{l,n}$, $l = 1, 2, \cdots, L_{1,n}$, to obtain

$$
\begin{aligned}
0 &= \int_{I_{l,n}} \left(\tilde{e}^{(m)}, \dot{\vartheta}^{(k)} + \nabla \cdot (\epsilon \nabla \vartheta^{(k)}) + \overline{f'(z^{(m)})}^{\top} \vartheta^{(k)} + \eta_R^{(k)} \right) dt \\
&= \left(\tilde{e}_{l,n}^{(m)-}, \vartheta_{l,n}^{(k)} \right) - \left(\tilde{e}_{l-1,n}^{(m)+}, \vartheta_{l-1,n}^{(k)} \right) + \int_{I_{l,n}} \left(-\dot{\tilde{e}}^{(m)} + f(u^{(m)}) - f(U^{(m)}), \vartheta^{(k)} \right) dt \\
&\quad + \int_{I_{l,n}} \left(\epsilon \nabla U^{(m)} - \epsilon \nabla u^{(m)}, \nabla \vartheta^{(k)} \right) dt + \int_{I_{l,n}} \left(\tilde{e}^{(m)}, \eta_R^{(k)} \right) dt.
\end{aligned}
$$

Hence,

$$
\begin{aligned}
0 &= \left(\tilde{e}_{l,n}^{(m)-}, \vartheta_{l,n}^{(k)} \right) - \left(\tilde{e}_{l-1,n}^{(m)-} - [U^{(m)}]_{l-1,n}, \vartheta_{l-1,n}^{(k)} \right) + \int_{I_{l,n}} \left(\dot{U}^{(m)} - f(U^{(m)}), \vartheta^{(k)} \right) dt \\
&\quad + \int_{I_{l,n}} \left(\epsilon \nabla U^{(m)}, \nabla \vartheta^{(k)} \right) dt + \int_{I_{l,n}} \left(\tilde{e}^{(m)}, \eta_R^{(k)} \right) dt \\
&\quad + \int_{I_{l,n}} \left(-\dot{u}^{(m)} + \nabla \cdot (\epsilon \nabla u^{(m)}) + f(u^{(m)}), \vartheta^{(k)} \right) dt
\end{aligned} \tag{22}
$$

Rearranging the terms in (22) and using (14) we obtain a recursive relation

$$
\begin{aligned}
\left(\tilde{e}_{l,n}^{(m)-}, \vartheta_{l,n}^{(k)} \right) &= \left(\tilde{e}_{l-1,n}^{(m)-}, \vartheta_{l-1,n}^{(k)} \right) - \left([U^{(m)}]_{l-1,n}, \vartheta_{l-1,n}^{(k)} \right) \\
&\quad - \int_{I_{l,n}} \left[\left(\dot{U}^{(m)} - f(U^{(m)}), \vartheta^{(k)} \right) + \left(\epsilon \nabla U^{(m)}, \nabla \vartheta^{(k)} \right) \right] dt \\
&\quad + \int_{I_{l,n}} \left(\delta_R^{(m)}, \vartheta^{(k)} \right) dt - \int_{I_{l,n}} \left(\tilde{e}^{(m)}, \eta_R^{(k)} \right) dt.
\end{aligned} \tag{23}
$$

This is the basis for the equation for the error at time t_n stated in the following lemma.

Lemma 4.2. *The iterative multi-discretization finite element solution satisfies an error equation over one time step:*

$$
\begin{aligned}
(\tilde{e}_n^{(m)-}, \psi_n) &= (\tilde{e}_{n-1}^{(m)-}, \vartheta_{n-1}^{(k)}) + \hat{Q}_{1,n} + \hat{Q}_{2,n} - \sum_{l=1}^{L_{1,n}} \int_{I_{l,n}} \left(\tilde{e}^{(m)}, \eta_R^{(k)} \right) dt \\
&\quad + \sum_{l=1}^{L_{1,n}} \int_{I_{l,n}} \left(\delta_R(u^{(m)}) - \delta_R(U^{(m)}), \vartheta^{(k)} \right) dt
\end{aligned}
$$

where

$$
\begin{aligned}
\hat{Q}_{1,n} = \sum_{l=1}^{L_{1,n}} \Big\{ \int_{I_{l,n}} \Big[\Big(-\dot{U}_1^{(m)} + f_1(U_1^{(m)}, U_2^{(m-1)}), \vartheta_1^{(k)} \Big) - \left(\epsilon_1 \nabla U_1^{(m)}, \nabla \vartheta_1^{(k)} \right) \Big] dt \\
- \left([U_1^{(m)}]_{l-1,n}, \vartheta_{1,l-1,n}^{(k)} \right) \Big\},
\end{aligned} \tag{24}
$$

$$
\begin{aligned}
\hat{Q}_{2,n} = \sum_{l=1}^{L_{1,n}} \Big\{ \int_{I_{l,n}} \Big[\Big(-\dot{U}_2^{(m)} + f_2(U_1^{(m)}, U_2^{(m)}), \vartheta_2^{(k)} \Big) - \left(\epsilon_2 \nabla U_2^{(m)}, \nabla \vartheta_2^{(k)} \right) \Big] dt \\
- \left([U_2{}^{(m)}]_{l-1,n}, \vartheta_{2,l-1,n}^{(k)} \right) \Big\}.
\end{aligned} \tag{25}
$$

Proof. This is obtained by using the recursive relation (23) and applying integration by parts. □

We note that this equation reflects the error arising from the consistent finite element numerical discretization of the analytical iterative method. Similar to Lemma 4.1, this error contains the iteration residuals weighted by the adjoint $\vartheta^{(k)}$. The last term cannot be approximated easily since it involves the error $\tilde{e}^{(m)}$ weighted by the iteration residual in the adjoint computation. However, provided that an *a*

priori estimate on $\tilde{e}^{(m)}$ is available, we can control this term to be relatively small due the fact that the residual can be made as small as needed when the adjoint computation is driven to convergence.

We now collect all the results above and obtain an error representation of the finite element multi-scale iterative implicit method by setting $e^{(m)} = u - U^{(m)} = (u - u^{(m)}) + (u^{(m)} - U^{(m)}) = \hat{e}^{(m)} + \tilde{e}^{(m)}$.

Theorem 4.1. *Set $\psi_N = \psi$ and $\psi_{n-1} = \vartheta_{n-1}^{(K_n)}$ in Alg. 4 and $\psi_{n-1} = \varphi_{n-1}^{(K_n)}$ in Alg. 3, for $n = N, N-1, \cdots, 1$. Then the error of iterative multi-discretization finite element solution at final time $t_N = T$ can be expressed as*

$$(e_N^{(M_n)-}, \psi_N) = (u_N - U_N^{(M_N)-}, \psi) = \sum_{n=1}^{N} \left(\hat{Q}_{1,n} + \hat{Q}_{2,n} + \hat{Q}_{3,n} + \hat{Q}_{4,n} + \hat{Q}_{5,n} + \hat{Q}_{6,n} \right), \tag{26}$$

$\hat{Q}_{1,n}$ and $\hat{Q}_{2,n}$ are given in Lemma 4.2 with $m = M_n$ and

$$\begin{aligned}
\hat{Q}_{3,n} &= \sum_{l=1}^{L_{1,n}} \int_{I_{l,n}} \left(-\delta_R(U^{(M_n)}), \vartheta^{(K_n)} \right) dt \\
\hat{Q}_{4,n} &= \sum_{l=1}^{L_{1,n}} \int_{I_{l,n}} \left(\delta_R(u^{(M_n)}), \vartheta^{(K_n)} - \varphi^{(K_n)} \right) dt \\
\hat{Q}_{5,n} &= \left(u_{n-1}^{(M_n)}, \Delta\Phi_n \psi_n \right) + \int_{I_n} \left(u^{(M_n)}, \xi_R^{(K_n)} \right) dt \\
\hat{Q}_{6,n} &= (\hat{e}_{n-1}^{(M_n)}, \Delta\Phi_n \psi_n) - \sum_{l=1}^{L_{1,n}} \int_{I_{l,n}} \left(\tilde{e}^{(M_n)}, \eta_R^{(K_n)} \right) dt,
\end{aligned}$$

Proof. First we estimate the error over one time step. Combining Lemma 4.2 and with Lemma 4.1 we get,

$$(e_n^{(M_n)-}, \psi_n) = (\tilde{e}_{n-1}^{(M_n)-}, \vartheta_{n-1}^{(K_n)}) + (\hat{e}_{n-1}^{(M_n)}, \phi_{n-1}^{(K_n)}) + \hat{Q}_{1,n} + \hat{Q}_{2,n} + \hat{Q}_{3,n} + \hat{Q}_{4,n} + \hat{Q}_{5,n} + \hat{Q}_{6,n}. \tag{27}$$

We note that since $U_{n-1}^{(M_n)-} = U_{n-1}^{(M_{n-1})-}$ and $u_{n-1}^{(M_n)} = u_{n-1}^{(M_{n-1})}$ (see Alg. 1), we have $\tilde{e}_{n-1}^{(M_n)-} = \tilde{e}_{n-1}^{(M_{n-1})-}$ and $\hat{e}_{n-1}^{(M_n)} = \hat{e}_{n-1}^{(M_{n-1})}$. This yields a recursive relation in terms of $\tilde{e}^{(M_n)-}$ and $\hat{e}^{(M_n)}$ for the total error over one time step. The error at the final time is obtained from undoing this relation and assuming $\tilde{e}_0^{M_0-} = \hat{e}_0^{M_0} = 0$. □

The terms $\hat{Q}_{5,n}$ and $\hat{Q}_{6,n}$ are not easy to approximate. However, provided the discretization error and the iteration error are sufficiently small, $\hat{Q}_{5,n}$ and $\hat{Q}_{6,n}$ are asymptotically small compared with $\hat{Q}_{1,n}, \ldots, \hat{Q}_{4,n}$.

Theorem 4.2. *The terms $\sum_{n=1}^{N} \hat{Q}_{5,n}$ and $\sum_{n=1}^{N} \hat{Q}_{6,n}$ are asymptotically small compared with $\sum_{n=1}^{N} \hat{Q}_{1,n}, \ldots, \sum_{n=1}^{N} \hat{Q}_{4,n}$ in the limit of iteration errors $\|u^{(M_n)} - u\|_{L^\infty(I_n;L^2(\Omega))}$ and $\|\varphi^{(K_n)} - \varphi\|_{L^\infty(I_n;L^2(\Omega))}$ tending to zero for all n.*

Proof. Of the two, $\hat{Q}_{5,n}$ is more difficult to estimate. Let $\hat{\varphi}$ be the solution of

$$\begin{cases}
-\dot{\hat{\varphi}} - \nabla \cdot (\epsilon \nabla \hat{\varphi}) = \overline{f'(u^{(M_n)})}^{\mathsf{T}} \hat{\varphi}, & (x, t) \in \Omega \times (t_n, t_{n-1}], \\
\hat{\varphi}(x, t) = 0, & (x, t) \in \partial\Omega \times (t_n, t_{n-1}], \\
\hat{\varphi}(x, t_n) = \psi_n(x), & x \in \Omega.
\end{cases} \tag{28}$$

Notice that (28) and (11) differ only in terms of linearization point for $\overline{f'}$. Now we write

$$\varphi - \varphi^{(K_n)} = (\varphi - \hat{\varphi}) + (\hat{\varphi} - \varphi^{(K_n)}) = \alpha + \beta,$$

where α and β satisfy, respectively

$$-\dot{\alpha} - \nabla\cdot(\epsilon\nabla\alpha) = \overline{f'(u^{(M_n)})}^{\top}\alpha + \delta_{f'}^{\top}\varphi, \tag{29}$$

$$-\dot{\beta} - \nabla\cdot(\epsilon\nabla\beta) = \overline{f'(u^{(M_n)})}^{\top}\beta - \xi_R^{(K_n)}, \tag{30}$$

with zero initial and boundary conditions, and we designate the 2×2 matrix $\delta_{f'} = \overline{f'(u)} - \overline{f'(u^{(M_n)})}$. Multiplying (29) by α, and following by integration over $(t, t_n)\times\Omega$ yields

$$\begin{aligned}\|\alpha(t)\|^2 &\le \|\alpha(t)\|^2 + 2\int_t^{t_n}(\epsilon\nabla\alpha,\nabla\alpha)\,d\tau\\ &= 2\int_t^{t_n}\big(\alpha,\overline{f'(u^{(M_n)})}\alpha\big)\,d\tau + 2\int_t^{t_n}\big(\varphi,\delta_{f'}\alpha\big)\,d\tau\\ &\le 2L\int_t^{t_n}\|\alpha\|^2\,d\tau + 2\int_t^{t_n}\int_\Omega|\varphi||\delta_{f'}||\alpha|\,d\tau,\end{aligned} \tag{31}$$

where $\|\cdot\|$ is the norm in $L^2(\Omega)\times L^2(\Omega)$, and $|\cdot|$ is understood as the usual Euclidean vector norm for φ and α, or its corresponding matrix norm for $\delta_{f'}$. There is a constant $C_\varphi < \infty$ such that $\|\varphi_i\|_{L^\infty(t,t_n,L^2(\Omega))} < C_\varphi$, see for example [20]. We apply the Cauchy-Schwarz and arithmetic-geometric mean inequalities to the last term on the right hand side of (31) to get

$$\begin{aligned}\|\alpha(t)\|^2 &\le 2L\int_t^{t_n}\|\alpha\|^2\,d\tau + \int_t^{t_n}\|\delta_{f'}\|^2\,d\tau + C_\varphi^2\int_t^{t_n}\|\alpha\|^2\,d\tau\\ &= \int_t^{t_n}\|\delta_{f'}\|^2\,d\tau + (2L + C_\varphi^2)\int_t^{t_n}\|\alpha\|^2\,d\tau.\end{aligned} \tag{32}$$

Gronwall's inequality then implies

$$\|\alpha(t)\|^2 \le \exp\big((2L + C_\varphi^2)(t_n - t)\big)\int_t^{t_n}\|\delta_{f'}\|^2\,d\tau. \tag{33}$$

Similarly, we get

$$\|\beta(t)\|^2 \le \exp\big((2L)(t_n - t)\big)\int_t^{t_n}\|\xi_R^{(K_n)}\|^2\,d\tau. \tag{34}$$

Next, we multiply $u^{(M_n)}$ to (29) and (30), respectively, integrate each of them over I_n, and apply integrations by parts, and use (14) to get

$$\begin{aligned}(u_{n-1}^{(M_n)},\alpha_{n-1}) - \int_{I_n}\Big(u^{(M_n)},\delta_{f'}^{\top}\varphi\Big)\,dt &= \int_{I_n}(\alpha,\delta_R^{(M_n)})\,dt,\\ (u_{n-1}^{(M_n)},\beta_{n-1}) + \int_{I_n}\Big(u^{(M_n)},\xi_R^{(K_n)}\Big)\,dt &= \int_{I_n}(\beta,\delta_R^{(M_n)})\,dt,\end{aligned} \tag{35}$$

and thus

$$\begin{aligned}\hat{Q}_{5,n} &= \int_{I_n}\Big(u^{(M_n)},\delta_{f'}^{\top}\varphi\Big)\,dt + \int_{I_n}(\alpha,\delta_R^{(M_n)})\,dt + \int_{I_n}(\beta,\delta_R^{(M_n)})\,dt\\ &\le \int_{I_n}\big(\delta_{f'}u^{(M_n)},\varphi\big)\,dt + \frac{1}{2}\int_{I_n}(\|\alpha\|^2 + \|\beta\|^2)\,dt + \frac{1}{2}\int_{I_n}\|\delta_R^{(M_n)}\|^2\,dt\\ &\le \int_{I_n}\Big(\overline{f'(u)}u^{(M_n)} - f(u^{(M_n)}),\varphi\Big)\,dt + \frac{1}{2}\int_{I_n}\exp\big(C(t_n - t)\big)\int_t^{t_n}(\|\delta_{f'}\|^2 + \|\xi_R^{(K_n)}\|^2)\,d\tau dt + \frac{1}{2}\int_{I_n}\|\delta_R^{(M_n)}\|^2\,dt.\end{aligned} \tag{36}$$

Notice that the second term and third terms in (36) involve integration of the square of residuals. Thus these terms are asymptotically small compared to $\hat{Q}_{j,n}$, $j = 1, \cdots, 4$ as $u^{(M_n)}$ converges to u. Moreover, the first integral in (36) dominates the other terms. We show this term is asymptotically small compared to $\hat{Q}_{3,n}$ as $u^{(M_n)}$ converges to u.

First we bound the term $\hat{Q}_{3,n}$. From assumption A.1 we have,

$$\begin{aligned}\hat{Q}_{3,n} &= \sum_{l=1}^{L_{1,n}} \int_{I_{l,n}} \left(f_1(U_1^{(M_n)}, U_2^{(M_n)}) - f_1(U_1^{(M_n)}, U_2^{(M_n-1)}), \vartheta_1^{(K_n)} \right) dt \\ &\leq L \sum_{l=1}^{L_{1,n}} \int_{I_{l,n}} \| U_2^{(M_n)} - U_2^{(M_n-1)} \| \| \vartheta_1^{(K_n)} \| \, dt.\end{aligned} \tag{37}$$

Now,

$$\begin{aligned}&\sum_{l=1}^{L_{1,n}} \int_{I_{l,n}} \| U_2^{(M_n)} - U_2^{(M_n-1)} \| \, dt \\ &\qquad \leq \sum_{l=1}^{L_{1,n}} \int_{I_{l,n}} \| U_2^{(M_n)} - u_2^{(M_n)} \| dt + \sum_{l=1}^{L_{1,n}} \int_{I_n} \| u_2^{(M_n)} - u_2^{(M_n-1)} \| dt + \sum_{l=1}^{L_{1,n}} \int_{I_{l,n}} \| u_2^{(M_n-1)} - U_2^{(M_n-1)} \| dt.\end{aligned} \tag{38}$$

Now, the first and the third terms are discretization errors, and depend on the order of the numerical method, say $\rho(\Delta t, h)$ for some homogeneous function ρ. From (68) in the proof of Theorem 9.1, $\| u_2^{(M_n)} - u_2^{(M_n-1)} \| = O(\tau^{M_n})$, for some $\tau < 1$, and

$$\int_{I_n} \| u_2^{(M_n)} - u_2^{(M_n-1)} \| \, dt = O(\tau^{M_n+1}). \tag{39}$$

Combining this with (37) and (38), we have,

$$\hat{Q}_{3,n} = O(\rho(\Delta t, h) + \tau^{M_n+1}). \tag{40}$$

We now return to estimate the first term in (36). Noting that $\overline{f'(u^{(M_n)})} u^{(M_n)} = f(u^{(M_n)})$ and by the assumption that f' is Lipschitz continuous with constant L', we have,

$$\int_{I_n} \left(\overline{f'(u)} u^{(M_n)} - f(u^{M_n}), \varphi \right) dt = \int_{I_n} \left((\overline{f'(u)} - \overline{f'(u^{(M_n)})}) u^{(M_n)}, \varphi \right) dt \leq \int_{I_n} L' \| u - u^{(M_n)} \| \| \phi \| \, dt. \tag{41}$$

An analysis of the semigroup associated with the problem similar to that used in the Appendix to derive (68) yields $\| u - u^{(M_n)} \| = O(\tau^{M_n+1})$. Combining this with (41) and using appropriate scaling we have,

$$\int_{I_n} \left(\overline{f'(u)} u^{(M_n)} - f(u^{M_n}), \varphi \right) dt = O(\tau^{M_n+2}). \tag{42}$$

Hence, $\hat{Q}_{5,n}$ is asymptotically smaller than $\hat{Q}_{3,n}$ as $u^{(M_n)}$ converges to u.

Turning to $\hat{Q}_{6,n}$, we note that it is a sum of two terms. The first term is a product of iteration errors for the forward and adjoint problems, and is straightforward to bound as smaller than $\hat{Q}_{j,n}$, $j = 1, \cdots, 4$ as the iterations converge. The second term in $\hat{Q}_{j,n}$, $j = 1, \cdots, 4$ is a product of discretization error and iteration residual in the adjoint. This is bounded smaller than $\hat{Q}_{j,n}$, $j = 1, o, 4$ by an argument similar to that used for analogous expressions in $\hat{Q}_{5,n}$.

□

4.3. A computational error estimate

The error representation in Theorem 4.1 contains terms involving the true continuum solution $u^{(M_n)}$ as well as the true adjoint solutions $\varphi^{(K_n)}$ and $\vartheta^{(K_n)}$. We form a computational error estimate by approximating the adjoint solutions, $\varphi^{(K_n),h}$ and $\vartheta^{(K_n),h}$, in a finite dimensional space. These adjoint problems are approximated by substituting the finite element solution $U^{(M_n)}$ for $u^{(M_n)}$, as is common in adjoint based error estimation literature. Further, the term $\hat{Q}_{4,n}$ is expressed as,

$$\hat{Q}_{4,n} = \sum_{l=1}^{L_{1,n}} \int_{I_{l,n}} \left(\delta_R(U^{(M_n)}), \vartheta^{(K_n)} - \varphi^{(K_n)}\right) dt + \left(\delta_R(u^{(M_n)}) - \delta_R(U^{(M_n)}), \vartheta^{(K_n)} - \varphi^{(K_n)}\right) dt \tag{43}$$

Here the term $\left(\delta_R(u^{(M_n)}) - \delta_R(U^{(M_n)}), \vartheta^{(K_n)} - \varphi^{(K_n)}\right)$ is a product of difference of two residuals, and hence we drop it in the computational error estimate. This leads to the following computational error estimate.

Theorem 4.3. *The error of the iterative multi-discretization finite element solution at final time $t_N = T$ can be approximated as,*

$$(e_N^{(M_n)-}, \psi_N) = (u_N - U_N^{(M_N)-}, \psi) \approx \sum_{n=1}^{N} \left(Q_{1,n} + Q_{2,n} + Q_{3,n} + Q_{4,n}\right), \tag{44}$$

where,

$$\begin{aligned}
Q_{1,n} &= \sum_{l=1}^{L_{1,n}} \Big\{ \int_{I_{l,n}} \Big[\Big(-\dot{U}_1^{(m)} + f_1(U_1^{(m)}, U_2^{(m-1)}), \vartheta_1^{(K_N),h} \Big) - \Big(\epsilon_1 \nabla U_1^{(m)}, \nabla \vartheta_1^{(K_n),h} \Big) \Big] dt \\
&\qquad - \Big([U_1^{(m)}]_{l-1,n}, \vartheta_{1,l-1,n}^{(K_n),h} \Big) \Big\}, \\
Q_{2,n} &= \sum_{l=1}^{L_{1,n}} \Big\{ \int_{I_{l,n}} \Big[\Big(-\dot{U}_2^{(m)} + f_2(U_1^{(m)}, U_2^{(m)}), \vartheta_2^{(K_n),h} \Big) - \Big(\epsilon_2 \nabla U_2^{(m)}, \nabla \vartheta_2^{(K_n),h} \Big) \Big] dt \\
&\qquad - \Big([U_2^{(m)}]_{l-1,n}, \vartheta_{2,l-1,n}^{(K_n),h} \Big) \Big\}, \\
Q_{3,n} &= \sum_{l=1}^{L_{1,n}} \int_{I_{l,n}} \left(-\delta_R(U^{(M_n)}), \vartheta^{(K_n),h} \right) dt \\
Q_{4,n} &= \sum_{l=1}^{L_{1,n}} \int_{I_{l,n}} \left(\delta_R(U^{(M_n)}), \vartheta^{(K_n),h} - \varphi^{(K_n),h} \right) dt
\end{aligned}$$

We present interpretations of the computational error contributions in Table 1. Note that we have

Notation	Contribution
Q_1	Discretization error in component U_1
Q_2	Discretization error in component U_2
Q_3	Iteration error for the numerical solution
Q_4	Error due to linearization in the computed adjoint problem

Table 1: Error contributions and their interpretations

dropped $\hat{Q}_{5,n}$ and $\hat{Q}_{6,n}$ to obtain (44). As explained, this is reasonable provided the iteration has converged to a sufficient degree and the discretization is sufficiently refined. The examples below demonstrate the estimate (44) provides a reasonably accurate approximation of the true error.

Remark 4.1. *We note that computing the error estimate* (44) *involves the cost of solving the adjoint problem in addition to computing the original approximation. The computational cost depends on how the numerical adjoint problem is solved, however the adjoint problem is at least linear, and hence often involves less iteration than solving the original problem.*

On this issue, it is important to note that if the practical application requires an accurate error estimate to accompany a numerical solution, then the issue of cost of the error estimate has to be related to the cost of alternative approaches to error estimation. There are other ways to treat numerical solutions of coupled systems involving iteration, e.g. [18, 17, 16, 7]. Some of these approaches provide for direct estimation of the effect of finite iteration on accuracy, at the cost of greatly increasing the number of adjoint problems that must be solved. The estimate in Theorem 4.1 is thus relatively inexpensive at the cost of assuming that the iteration has converged to a sufficient degree.

Remark 4.2. *Standard adaptive error control strategies based on the Principle of Equidistribution applied to "dual-weighted"* a posteriori *estimates, [8, 9, 5, 19, 3], can be extended to* (44) *in a straightforward way to balance all sources of error. For example, if the component Q_1 is large, then refining the spatial and temporal mesh for the first component may lead to a more accurate solutions. A similar conclusion follows for Q_2. The terms Q_3 and Q_4 reflect errors incurred due to finite iterations, and these errors may be reduced by increasing the number of iterations. However, we note that many application codes for multi-physics problems eschew adaptive computation.*

5. Numerical experiments using equal spatial meshes

In this section, we present numerical examples to illustrate the performance of the error estimates. For various problems, we show plots of the error estimate and true error accompanied by plots of the individual contributions to the error estimate, $Q_{1,n}, Q_{2,n}, Q_{3,n}, Q_{4,n}$ as defined in Lemma 4.2 and Theorem 4.3. A comparison of the relative sizes of the different contributions to the error is often illuminating.

All forward problems are solved using continuous piecewise linear functions in space and using the piecewise constant discontinuous Galerkin method in time. The piecewise constant discontinuous Galerkin method, or dG(0), is equivalent to the backward Euler scheme. The nonlinear equations are solved using Newton's Method. The adjoint solutions are approximated using continuous piecewise quadratic functions in space and piecewise linear continuous Galerkin method in time. The piecewise linear continuous Galerkin method, or cG(1), is equivalent to the second order Crank-Nicholson scheme. All problems are posed on the unit square, i.e., on $\Omega = [0,1] \times [0,1]$ and solved using a uniform mesh containing $(20 \times 20 \times 2)$ triangular elements. The initial conditions at time $t = 0$ are $u = (\sin(\pi x_1)\sin(\pi x_2), \sin(\pi x_1)\sin(\pi x_2))^T$.

The quantity of interest in all cases is given by the globally supported function $\psi = (\sin(\pi x_1)\sin(\pi x_2), \sin(\pi x_1)\sin(\pi x_2))^T$. We compare the performance of estimators using either the analytical solution when available. Otherwise we use a "reference solution" using a higher order spatial discretization and a finer time step. In our numerical results, we plot the different error components and tabulate the effectivity ratio of the estimator. The effectivity ratio is defined as the ratio of the estimated error to the true error in the quantity of interest, provided the true error is not zero. An accurate error estimator has effectivity ratio close to one.

5.1. An equal rate one-way coupled linear system

We consider the system,

$$\begin{cases} \dot{u}_1 - \Delta u_1 = \pi^2 u_1, \\ \dot{u}_2 - \Delta u_2 = \pi^2 (0.5 u_2 + u_1). \end{cases}$$

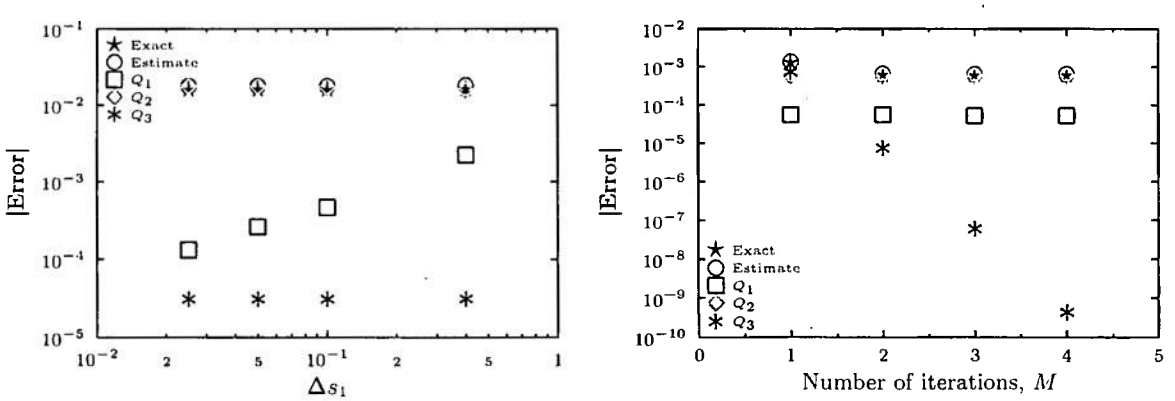
Notice that this is a one-way coupled system in which the variable of subsystem 1, u_1, is coupled to the variable of subsystem 2, but u_1 can be solved independently of u_2. The exact solution is $u_1 = e^{-\pi^2 t}\sin(\pi x_1)\sin(\pi x_2)$ and $u_2 = 2e^{-\pi^2 t}\sin(\pi x_1)\sin(\pi x_2)$, hence there is not a significant difference in spatial or temporal scales. Since the system is only coupled in one direction there is no need to iterate to solve the system and there is no iteration error, i.e., $Q_3 = 0$. Moreover, for linearly coupled systems $\phi = \vartheta$, and hence $Q_4 = 0$. The system is solved until $T = 0.2$ with $\Delta t = \Delta s_1 = \Delta s_2 = 0.02$. The error estimate was -0.0177161, as compared to the true error of -0.0169774 for an effectivity ratio of 1.04.

5.2. A multirate coupled linear system

We consider the system,

$$\begin{cases} \dot{u}_1 - \Delta u_1 = -1000u_1 + u_2 \\ \dot{u}_2 - \Delta u_2 = 999u_1 - 2u_2. \end{cases} \tag{45}$$

Here, u_1 is a fast variable and u_2 is a slow variable. We solve until $T = 0.2$ and plot the error components as a function of Δs_1 in Fig. 3(a) while fixing $\Delta t = \Delta s_2 = 0.4$. We use two iterations at each of the time steps Δt. As expected, the error in the component Q_1 decreases as Δs_1 is reduced. In Fig. 3(b) we plot the effect of employing different number of iterations to solve the system at each time step Δt. In this case, we use $\Delta t = 0.04$, $\Delta s_1 = \Delta t/32$ and $\Delta s_2 = \Delta t/16$. The iteration error decreases as the number of iterations is increased. Except in the extreme case of just one iteration, the contribution to the error from iteration is relatively small. In all cases, the error estimator provided an accurate prediction of the exact error. We recall that for linear systems, $\phi = \vartheta$, and hence $Q_4 = 0$. The accuracy of the estimator is also illustrated in Tables 2, which show effectivity ratios close to the ideal value of 1.0.



(a) (b)

Figure 3: Example 5.2: $T = 0.2$. (a) $\Delta t = \Delta s_2 = 0.4$, $M = 2$. Error contributions as Δs_1 is varied. (b) $\Delta t = 0.04$, $\Delta s_1 = \Delta t/32$, $\Delta s_2 = \Delta t/16$. Error contributions as M is varied.

5.3. A coupled nonlinear system using different time steps

We consider the system,

$$\begin{cases} \dot{u}_1 - \Delta u_1 = u_1^4 + u_2^2, \\ \dot{u}_2 - \Delta u_2 = u_1 - u_2^3. \end{cases} \tag{46}$$

The system is solved until $T = 0.2$, with $\Delta t = 0.04$, $\Delta s_1 = \Delta t/16$ and $\Delta s_2 = \Delta t/2$. In Fig. 4 the result of increasing the number of iterations is demonstrated. The component Q_3 is initially large, but decays to

Δs_1	Effectivity Ratio
0.4	1.12
0.1	1.11
0.05	1.12
0.025	1.12

(a)

M	Effectivity Ratio
1	1.06
2	1.09
3	1.09
4	1.09

(b)

Table 2: Effectivity Ratios for the experiment in Fig. 3. a) Effectivity Ratios as Δs_1 is varied. (b) Effectivity Ratios as M is varied.

a small value after two iterations. The component Q_4 is nonzero for this problem, since the adjoints ϑ and ϕ differ from one another. However, it is quite small compared to other components. Again, we obtained very accurate error estimates. Once again, the effectivity ratios, shown in Table 3 are close to the ideal value of 1.0.

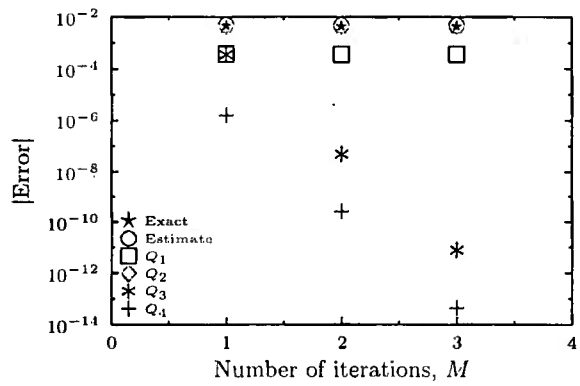


Figure 4: Example 5.3: $T = 0.2$, $\Delta t = 0.04$, $\Delta s_1 = \Delta t/16$, $\Delta s_2 = \Delta t/2$. Error contributions as the number of iterations M is varied.

M	Effectivity Ratio
1	1.08
2	1.09
3	1.09

Table 3: Effectivity Ratios for the experiment in Fig. 4.

5.4. A coupled nonlinear system using equal time steps

We consider the system,

$$\begin{cases} \dot{u}_1 - \Delta u_1 = \quad \exp(u_1) + \exp(u_2) - 2, \\ \dot{u}_2 - \Delta u_2 = -\exp(u_1) - \exp(u_2) + 2. \end{cases} \tag{47}$$

The system is solved until $T = 0.2$, with $\Delta t = 0.01$, $\Delta s_1 = \Delta s_2 = \Delta t/2$. The effect of increasing the number of iterations is shown in Fig. 5. The component Q_3 is large after just one iteration, but contributes

relatively little after two iterations. The component Q_4 is nonzero for this problem since the adjoints ϑ and ϕ differ from one another. The effectivity ratios for this experiment are shown in Table 4. The effectivity ratios are quite close to 1.0, indicating the accuracy of our estimator.

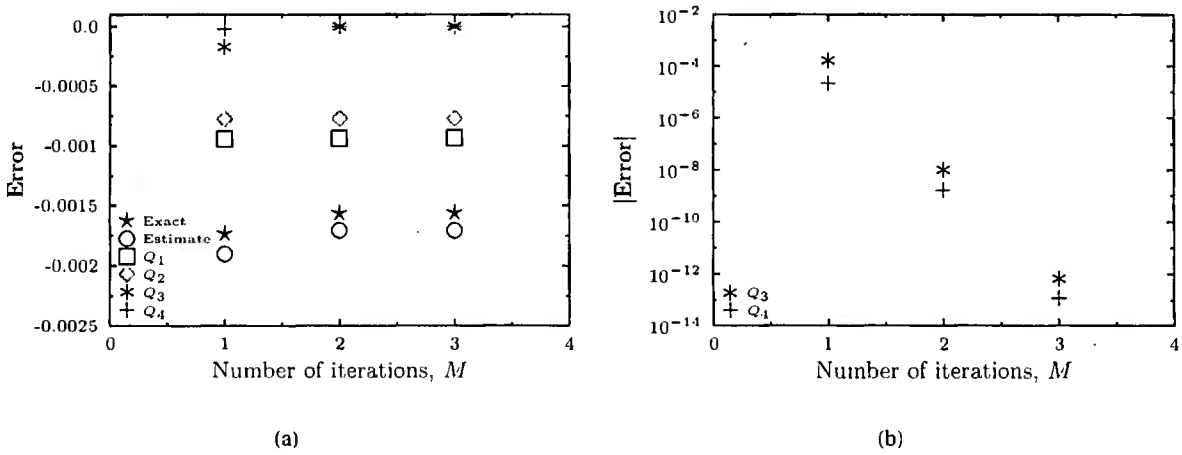


Figure 5: Example 5.4: $T = 0.2$, $\Delta t = 0.01$, $\Delta s_1 = \Delta s_2 = \Delta t/2$. Error contributions as the number of iterations M is varied. (a) True and estimated errors. (b) Q_3 and Q_4 only.

M	Effectivity Ratio
1	1.10
2	1.09
3	1.09

Table 4: Effectivity Ratios for the experiment in Fig. 5 (a).

6. *A posteriori* analysis of the iterative multi-discretization Galerkin finite element method for different spatial meshes

In this section, we derive an estimate for the case in which the two subsystems in Alg. 2 are solved on different space meshes. For such systems, we can further decompose the error components to reflect the projection errors. Solution of (4) involves the projection of $U_2^{(m-1)}$, denoted as $\Pi_{2\to 1}U_2^{(m-1)}$, from $W_{l,n}^{q_1}$ to $W_{k,n}^{q_2}$. If the number of time steps are the same for the two subsystems, then $\Pi_{2\to 1}$ is the projection of functions from the mesh for subsystem 1 to functions on the mesh for subsystem 2. Similarly, solution to (5) involves the projection, $\Pi_{1\to 2}U_1^{(m)}$, of $U_1^{(m)}$ on the space of functions on the mesh of subsystem 2. With these projections we have the following error representation.

Theorem 6.1. *Set $\psi_N = \psi$ and $\psi_{n-1} = \vartheta_{n-1}^{K_n}$ for $n = N, N-1, \cdots, 1$. Then, with Assumptions A.1 and A.2, the error of iterative multi-discretization finite element solution at final time $t_N = T$ can be expressed as*

$$(u_N - U_N^{(M_N)-}, \psi) = \sum_{n=1}^{N} \left(\hat{Q}_{1b,n} + \hat{Q}_{1c,n} + \hat{Q}_{2b,n} + \hat{Q}_{2c,n} + \hat{Q}_{3,n} + \hat{Q}_{4,n} + \hat{Q}_{5,n} + \hat{Q}_{6,n} \right), \tag{48}$$

where $\hat{Q}_{3,n}$, $\hat{Q}_{4,n}$, $\hat{Q}_{5,n}$, $\hat{Q}_{6,n}$ *are as given in Theorem 4.3 and*

$$\begin{aligned}
\hat{Q}_{1b,n} &= \sum_{l=1}^{L_{1,n}} \Big\{ \int_{I_{l,n}} \Big[\Big(-\dot{U}_1^{(m)} + f_1(U_1^{(m)}, \Pi_{2\to 1} U_2^{(m-1)}), \vartheta_1^{(k)} \Big) - \Big(\epsilon_1 \nabla U_1^{(m)}, \nabla \vartheta_1^{(k)} \Big) \Big] dt \\
&\qquad + \Big([U_1^{(m)}]_{l-1,n}, \vartheta_{1,l-1,n}^{(k)} \Big) \Big\}, \\
\hat{Q}_{1c,n} &= \Big(f_1(U_1^{(m)}, U_2^{(m-1)}) - f_1(U_1^{(m)}, \Pi_{2\to 1} U_2^{(m-1)}), \vartheta_1^{(k)} \Big) \\
\hat{Q}_{2b,n} &= \sum_{l=1}^{L_{1,n}} \Big\{ \int_{I_{l,n}} \Big[\Big(-\dot{U}_2^{(m)} + f_2(\Pi_{1\to 2} U_1^{(m)}, U_2^{(m)}), \vartheta_2^{(k)} \Big) - \Big(\epsilon_2 \nabla U_1^{(m)}, \nabla \vartheta_1^{(k)} \Big) \Big] dt \\
&\qquad + \Big([U_2{}^{(m)}]_{l-1,n}, \vartheta_{2,l-1,n}^{(k)} \Big) \Big\}, \\
\hat{Q}_{2c,n} &= \Big(f_2(U_1^{(m)}, U_2^{(m)}) - f_2(\Pi_{1\to 2} U_1^{(m)}, U_2^{(m)}), \vartheta_2^{(k)} \Big).
\end{aligned}$$

Proof. Adding and subtracting $\Big(f_1(U_1^{(m)}, \Pi_{2\to 1} U_2^{(m-1)}), \vartheta_1^{(k)} \Big)$ to (24),

$$\begin{aligned}
\hat{Q}_{1,n} &= \sum_{l=1}^{L_{1,n}} \Big\{ \int_{I_{l,n}} \Big[\Big(-\dot{U}_1^{(m)} + f_1(U_1^{(m)}, \Pi_{2\to 1} U_2^{(m-1)}), \vartheta_1^{(k)} \Big) - \Big(\epsilon_1 \nabla U_1^{(m)}, \nabla \vartheta_1^{(k)} \Big) \Big] dt \\
&\qquad + \Big([U_1^{(m)}]_{l-1,n}, \vartheta_{1,l-1,n}^{(k)} \Big) \\
&\qquad + \Big(f_1(U_1^{(m)}, U_2^{(m-1)}) - f_1(U_1^{(m)}, \Pi_{2\to 1} U_2^{(m-1)}), \vartheta_1^{(k)} \Big) \Big\} \\
&= \hat{Q}_{1b,n} + \hat{Q}_{1c,n}
\end{aligned}$$

Similarly, adding and subtracting $\Big(f_1(\Pi_{1\to 2} U_1^{(m)}, U_2^{(m)}), \vartheta_1^{(k)} \Big)$ to (25) leads to,

$$\hat{Q}_{2,n} = \hat{Q}_{2b,n} + \hat{Q}_{2c,n}$$

Combining these with (26) leads to (48). □

For simplicity in our examples, one mesh will always be a refinement of the other mesh. Nodal projection for the space meshes is employed for the operators $\Pi_{1\to 2}$ and $\Pi_{2\to 1}$. Further, we form a computational error estimate in the manner outlined in Section 4.3, representing the approximations of the terms $\hat{Q}_{i,n}$ as $Q_{i,n}$. We recall Table 1 that describes the contributions to the error.

6.1. A Linear System

In this section, we consider the system,

$$\begin{aligned}
&\dot{u}_1 - \Delta u_1 = u_1, && (\boldsymbol{x}, t) \in \Omega \times (0, T], \\
&\dot{u}_2 - \Delta u_2 = \mathbf{b} \cdot \nabla u_1 + u_2, && (\boldsymbol{x}, t) \in \Omega \times (0, T], \\
&u_1 = 5x_1^2(1-x_1)^2(e^{10x_1^2} - 1)x_2^2(1-x_2)^2(e^{10x_2^2} - 1), && (\boldsymbol{x}, t) \in \Omega \times \{0\}, \\
&u_2 = \sin(\pi x_1)\sin(\pi x_2), && (\boldsymbol{x}, t) \in \Omega \times \{0\},
\end{aligned} \tag{49}$$

where $\mathbf{b} = (1000, 1000)^\mathsf{T}$. The quantity of interest is taken to be $\psi = (0, 100x_1^2(1-x_1)^2x_2^2(1-x_2)^2)^\mathsf{T}$. Note that due to the presence of the term $\mathbf{b}\cdot\nabla u_1$, the term $\overline{f_{21}'(u^{(m)})}\phi_2^{(k)}$ in Alg. 3 is interpreted as $-\nabla\cdot(\mathbf{b}\phi_2^{(k)})$. The term $\overline{f_{21}'(z^{(m)})}\vartheta_2^{(k)}$ in Alg. 4 is treated in a similar fashion.

In the numerical experiments, subsystem 1 is solved on a uniform mesh comprising $(40 \times 40 \times 2)$ triangular elements. The mesh for subsystem 2 is varied through $(5 \times 5 \times 2)$, $(10 \times 10 \times 2)$, $(20 \times 20 \times 2)$ and

$(40 \times 40 \times 2)$ triangular elements and the system is solved with $\Delta t = \Delta s_1 = \Delta s_2 = 0.01$. We plot the error components as a function of the ratio of mesh sizes in Fig. 6. The figure indicates that the projection error Q_{2c} dominates the total error when there is a large difference between the mesh sizes, and goes to 0 as the two meshes have the same size.

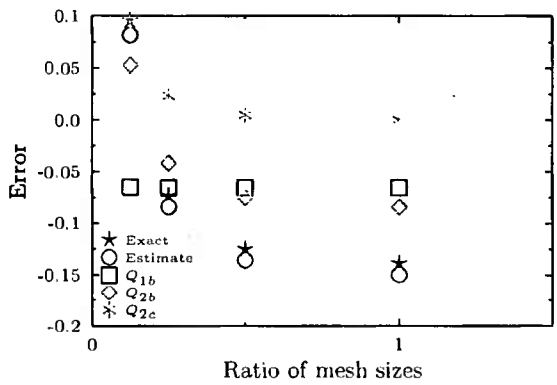


Figure 6: Example 6.1: $T = 0.2$, $\Delta t = \Delta s_1 = \Delta s_2 = 0.01$. Error contributions versus ratio of mesh sizes.

6.2. The Brusselator

We recall the Brusselator problem (1) in Sec. 1. The values of different parameters are the same as in Section 1. The system as posed does not satisfy $f(\tilde{u}) = 0$. However, we use a change of variable to accomplish this. The new variables are defined as,

$$\begin{cases} u_1 = \tilde{u}_1 - \alpha \\ u_2 = \tilde{u}_2 - \beta/\alpha \end{cases}$$

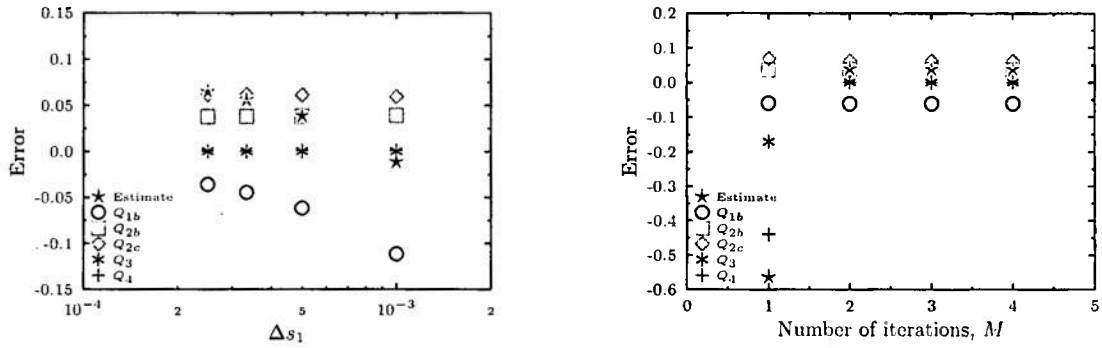
With these new variables, $u = (u_1, u_2)^\top$, the new set of equations satisfy the requirement that $f(u) = 0$. We experiment with two different quantities of interest; a spatial quantity of interest at the final time, and a time based quantity of interest approximating the temporal derivative at a certain time.

6.2.1. A spatial quantity of interest at the final time

For this experiment, we take the quantity of interest to be

$$\psi = \begin{pmatrix} x_1^2(1-x_1)^2(\exp(6x_1^2)-1)x_2^2(1-x_2)^2(\exp(6x_2^2)-1) \\ 0 \end{pmatrix},$$

evaluated at final time $T = 0.7$. This quantity of interest is adapted from Ch. 8 in [2]. Mesh 1 and mesh 2 were chosen to be uniform with $(40 \times 40 \times 2)$ and $(20 \times 20 \times 2)$ triangular elements respectively, $\Delta t = \Delta s_2 = 0.001$ and $M = 2$. In Fig. 7(a) the effect of decreasing Δs_1 on the error components is evident and the error component Q_{1b} decreases as Δs_1 is reduced as expected. Note that the total error increases as Δs_1 is reduced, due to cancellation of errors with opposite sign. In Fig. 7(b) the effect of varying the number of iterations M is shown. The rest of the parameters are the same as for Fig. 7(a) except that Δs_1 is fixed at 0.001. For $M = 1$ there are significant errors in the components Q_3 and Q_4, but these errors decrease as the number of iterations is increased.



(a) (b)

Figure 7: Brusselator: $T = 0.7$, $\Delta t = \Delta s_2 = 0.001$. (a) $M = 2$. Error contributions as Δs_1 is varied. (b) $\Delta s_1 = 0.001$. Error contributions as M is varied.

6.2.2. Competing effects of discretization and projection

Separate refinement of either of the spatial meshes may result in a reduction of discretization errors for the solution component(s) computed on that mesh, but may also increase projection errors. For this experiment, mesh 2 was held fixed with $(20 \times 20 \times 2)$ triangular elements while mesh 1 was varied having $(20 \times 20 \times 2)$, $(40 \times 40 \times 2)$ and $(80 \times 80 \times 2)$ triangular elements. Here $\Delta t = \Delta s_1 = \Delta s_2 = 0.001$ and $M = 2$. In Fig. 8(a) the error components are plotted for this series of different discretization levels for mesh 1. Note that discretization error Q_{1b} decreased as the mesh ratio decreased (as mesh 1 was refined), but that the projection error Q_{2c} *increased*. While the magnitude of reduction of Q_{1b} exceeded the magnitude of the increase in Q_{2c}, the *total* error *increased* as mesh 1 was refined due to cancellation of errors with opposite sign.

In Table 5 we tabulate the error contributions for three different choices of mesh 1, two uniform and one non-uniform. Fig. 8(b) where the mesh is refined in regions of rapid variation of component u_1. Mesh 2 was uniform with $(20 \times 20 \times 2)$ triangular elements for all three cases. Here $\Delta t = \Delta s_2 = 0.001$, $\Delta s_1 = \Delta t/4 = 0.00025$, and $M = 2$.

When mesh 1 has $(20 \times 20 \times 2)$ uniform triangular elements, the first row of Table 5 indicates that the dominant error contribution is Q_{1b}, the discretization error on mesh 1. Halving each element on mesh 1 produces a situation in which the discretization errors on both meshes and the projection error are roughly of the same magnitude (row 2 of Table 5). Non-uniform refinement of mesh 1 such that it has a finer mesh in regions of sharp variation produces a similar distribution of error with 2/3 of the number of elements (row 3 of Table 5).

Mesh	Elements	Dof_1	Dof_2	Estimate	Q_{1b}	Q_{2b}	Q_{2c}	Q_3	Q_4
Coarse uniform	800	441	441	-0.1509	-0.2139	0.0618	0.0000	0.0003	0.0008
Fine uniform	3200	1681	441	0.0644	-0.0358	0.0377	0.0614	0.0002	0.0007
Non-uniform	2320	1241	441	0.0600	-0.0511	0.0398	0.0704	0.0003	0.0007

Table 5: Brusselator: $T = 0.7$, $\Delta t = \Delta s_2 = 0.001$, $\Delta s_1 = \Delta t/4 = 0.00025$, $M = 2$. Error components for two uniform and one non-uniform mesh 1. Here Dof_i refers to degrees-of-freedom for the component u_i.

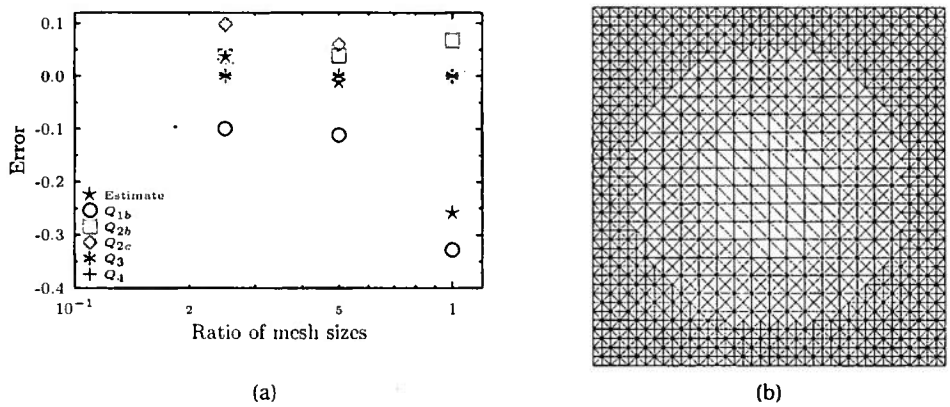


Figure 8: Brusselator: $T = 0.7, \Delta t = \Delta s_1 = \Delta s_2 = 0.001, M = 2$. (a) Error contributions versus ratio of mesh sizes. (b) Refined mesh used to produce error contributions provided in row 3 of Table 5.

6.2.3. A temporal derivative as the quantity of interest

For this experiment, we approximate the time derivative $\frac{\partial}{\partial t}\int_\Omega u_1\,dx$ of the average value of u_1 at some $t = t_D$ using a central difference. We approximate the temporal derivative of a function v by,

$$\left.\frac{\partial v}{\partial t}\right|_{t=t_D} \approx \frac{v(t_D+0.5\Delta t)-v(t_D-0.5\Delta t)}{\Delta t}. \tag{50}$$

In practice we approximate the point value using a local average. That is, $v(\tau) \approx \overline{v(\tau)} = \int_{\tau-r}^{\tau+r} v(t)\,dt$. As $r \to 0$, $\overline{v(\tau)} \to v(\tau)$. The adjoint solution required a finer (time) discretization near t_D to accurately resolve the adjoint solution. Near $t = t_D$, we used a time discretization that was 100 times finer than that used for the forward problem. That is, in this region the time step is $\Delta t/100$, where Δt is the time step for the forward problem. Moreover, we chose $r = \Delta t/10$.

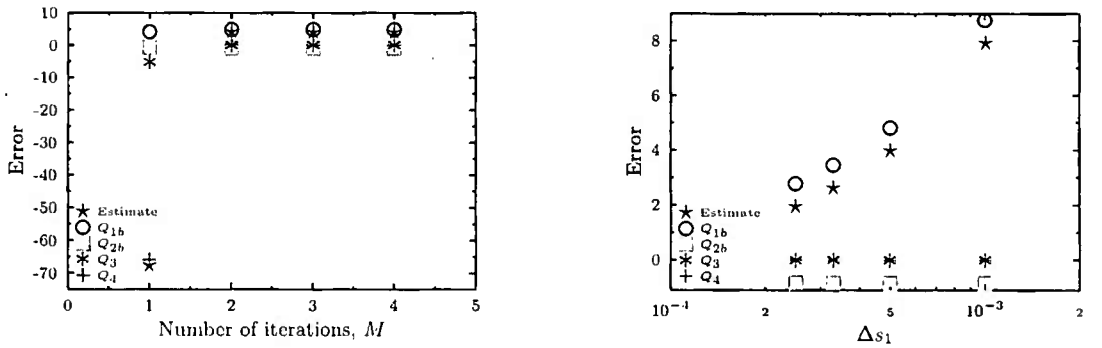
In Fig. 9(a) we investigate the effect of the number of iterations. We use $\Delta t = \Delta s_2 = 0.001$, $\Delta s_1 = 0.0005$ and the same uniform mesh with $(40 \times 40 \times 2)$ triangular elements for both components. For $M = 1$, the estimate is dominated by the term Q_3 and then by Q_4, which measure the effect of the number of iterations.

In Fig. 9(b) we show the effect of varying Δs_1 for fixed M. We use $\Delta t = \Delta s_2 = 0.001$, $M = 3$ and the same uniform mesh with $(40 \times 40 \times 2)$ triangular elements for both components. We see that the error in the component Q_{1b} decreases as Δs_1 is reduced, as expected. The component Q_{1b} also dominates the total error, so refining the time steps for this fast component leads to significant reduction of total error as well.

7. *A posteriori* analysis for systems with nonlinear diffusion coefficients

To explain how the *a posteriori* analysis can be extended to fully nonlinear coupled systems, we provide a formal derivation of an *a posteriori* error estimates for systems of parabolic initial boundary value problems having nonlinear diffusion coefficients. We consider the problem of finding $u = (u_1, u_2)^\top$ for systems in which the diffusion coefficient, $\epsilon = \epsilon(u)$ is a function of u,

$$\begin{cases} \dot{u}_1 - \nabla\cdot\left(\epsilon_1(u_1,u_2)\nabla u_1\right) = f_1(u_1,u_2), & (\boldsymbol{x},t)\in\Omega\times(0,T], \\ \dot{u}_2 - \nabla\cdot(\epsilon_2(u_1,u_2)\nabla u_2) = f_2(u_1,u_2), & (\boldsymbol{x},t)\in\Omega\times(0,T], \\ u_i(\boldsymbol{x},t) = 0, & (\boldsymbol{x},t)\in\partial\Omega\times(0,T], i=1,2 \\ u_i(\boldsymbol{x},0) = g_i(\boldsymbol{x}), & \boldsymbol{x}\in\Omega, i=1,2, \end{cases} \tag{51}$$



(a) (b)

Figure 9: Brusselator: Time derivative at $t_D = 0.7$ (final time is $T = 0.8$). (a) Effect as the number of iterations M varies. (b) Effect as Δs_1 varies given $M = 3$.

or in a compact form,

$$\dot{u} - \nabla \cdot (\epsilon(u) \nabla u) = f(u),$$

where $\epsilon(u) = \text{diag}(\epsilon_1(u) \;\; \epsilon_2(u))$. Meaningful analysis of general parabolic systems with nonlinear diffusion coefficients is very challenging, see for example [20]. Generally, analytic results can be greatly improved by employing the special properties of particular systems. We assume that the *a priori* analysis is in place and proceed to focus on the *a posteriori* analysis.

We again use Alg. 1 to solve this system, with the obvious modification that $\epsilon = \epsilon(u)$ and for simplicity we consider a scenario of having the same spatial discretization for u_1 and u_2. The adjoint problem similar to (11) is modified to account for the dependence of ϵ on u,

$$-\dot{\varphi} - \nabla \cdot (\overline{\epsilon(u)} \nabla \varphi) + \overline{\epsilon'(u)}^{\mathsf{T}} \cdot \nabla \varphi = \overline{f'(u)}^{\mathsf{T}} \varphi, \tag{52}$$

where $\overline{\epsilon'(u)}$ denotes a square matrix and $\overline{\epsilon(u)}$ is a diagonal matrix whose entries, respectively, are

$$\overline{\epsilon'_{ij}(u)} = \left[\int_0^1 \frac{\partial \epsilon_i}{\partial u_j}(su) \nabla (u_i s)\, ds \right] \quad \text{and} \quad \overline{\epsilon_i(u)} = \int_0^1 \epsilon_i(us) ds. \tag{53}$$

In compact form, the adjoint problems similar to (17) is also modified as

$$-\dot{\varphi}^{(k)} - \nabla \cdot (\overline{\epsilon(u^{(m)})} \nabla \varphi^{(k)}) + \overline{\epsilon'(u^{(m)})}^{\mathsf{T}} \cdot \nabla \varphi^{(k)} = \overline{f'(u^{(m)})}^{\mathsf{T}} \varphi^{(k)} + \xi_R^{(k)} + \xi_D^{(k)}, \tag{54}$$

where $\xi_R^{(k)}$ is as defined in (17), $\overline{\epsilon'(u^{(m)})}$ and $\overline{\epsilon(u^{(m)})}$ are similarly defined as in (53), and

$$\xi_D^{(k)} = \left[0 \;\; \overline{\epsilon'_{12}(u^{(m)})}^{\mathsf{T}} \cdot \nabla (\varphi_1^{(k)} - \varphi_1^{(k-1)}) \right]^{\mathsf{T}}.$$

Similarly, the finite element adjoint problem (21) is modified as,

$$-\dot{\vartheta}^{(k)} - \nabla \cdot (\overline{\epsilon(z^{(m)})} \nabla \vartheta^{(k)}) + \overline{\epsilon'(z^{(m)})}^{\mathsf{T}} \cdot \nabla \vartheta^{(k)} = \overline{f'(z^{(m)})}^{\mathsf{T}} \vartheta^{(k)} + \eta_R^{(k)} + \eta_D^{(k)}, \tag{55}$$

where $z^{(m)} = su^{(m)} + (1-s)U^{(m)}$, $\overline{\epsilon'(z^{(m)})}$ is a square matrix and $\overline{\epsilon(z^{(m)})}$ is a diagonal matrix whose entries, respectively, are

$$\overline{\epsilon'_{ij}(z^{(m)})} = \int_0^1 \frac{\partial \epsilon_i}{\partial u_j}(z_i{}^{(m)}) \nabla z^{(m)} ds \quad \text{and} \quad \overline{\epsilon_i(z^{(m)})} = \int_0^1 \epsilon_i(z^{(m)}) ds. \tag{56}$$

The residual $\eta_R^{(k)}$ is as defined in (21) and

$$\eta_D^{(k)} = \left[0 \quad \overline{\epsilon'_{12}(z^{(m)})} \cdot \nabla(\vartheta_1^{(k)} - \vartheta_1^{(k-1)})\right]^\top.$$

Analysis of this system leads to the following error representation.

Theorem 7.1. *Set $\psi_N = \psi$ and $\psi_{n-1} = \vartheta_{n-1}^{K_n}$ for $n = N, N-1, \cdots, 1$. Then the error of iterative multi-discretization finite element solution of (51) at final time $t_N = T$ can be expressed as*

$$(u_N - U_N^{(M_N)-}, \psi) = \sum_{n=1}^{N} \left(\hat{Q}_{1,n} + \hat{Q}_{2,n} + \hat{Q}_{3,n} + \hat{Q}_{4,n} + \hat{Q}_{5b,n} + \hat{Q}_{6b,n} + \hat{Q}_{5c,n} + \hat{Q}_{6c,n}\right), \tag{57}$$

where $\hat{Q}_{1,n}$, $\hat{Q}_{2,n}$, $\hat{Q}_{3,n}$, $\hat{Q}_{4,n}$ are as given in Theorem 4.3, and

$$\hat{Q}_{5b,n} = \left(u_{n-1}^{(M_n)}, \Delta\Phi_n\psi_n\right) + \int_{I_n} \left(u^{(M_n)}, \xi_R^{(K_n)} + \xi_D^{(K_n)}\right) dt$$

$$\hat{Q}_{6b,n} = \left(\hat{e}_{n-1}, \Delta\Phi_n\psi_n\right) - \sum_{l=1}^{L_{1,n}} \int_{I_{l,n}} \left(\tilde{e}^{(M_n)}, \eta_R^{(K_n)} + \eta_D^{(K_n)}\right) dt$$

$$\hat{Q}_{5c,n} = \sum_{l=1}^{L_{1,n}} \int_{I_{l,n}} \left(\delta\epsilon_1(U^{(M_n)})\nabla U_1^{(M_n)}, \nabla\vartheta_1^{(K_n)}\right) dt$$

$$\hat{Q}_{6c,n} = \sum_{l=1}^{L_{1,n}} \int_{I_{l,n}} \left(\delta\epsilon_1(u^{(M_n)})\nabla u_1^{(M_n)}, \nabla\vartheta_1^{(K_n)} - \nabla\varphi_1^{(K_n)}\right) dt,$$

with $\delta\epsilon_1(u^{(m)}) = \epsilon_1(u_1^{(m)}, u_2^{(m)}) - \epsilon_1(u_1^{(m)}, u_2^{(m-1)})$, and similarly for $\delta\epsilon_1(U^{(m)})$.

A proof similar to that of Lemma 4.2 is beyond the scope of this paper. With the appropriate a priori analysis in place, we expect that the terms $\hat{Q}_{5b,n}, \hat{Q}_{5c,n}, \hat{Q}_{6b,n}$ and $\hat{Q}_{6c,n}$ are small compared to $\hat{Q}_{1,n} \cdots \hat{Q}_{4,n}$.

8. Conclusions

In this paper we formulate and analyze an iterative multi-discretization Galerkin finite element method for multi-scale reaction-diffusion equations. Subsystems in such reaction-diffusion equations may exhibit significantly different spatial and temporal scales, motivating a multi-discretization numerical method. We employ adjoint operators and variational analysis to form computational error estimates for a quantity of interest calculated from the multi-discretization finite element method. A key insight in analyzing the multi-discretization method is the realization that the adjoint operator associated with the iterative multi-discretization approximation is different from that of the original problem. Hence, our analysis utilizes two adjoint operators. One of the operators utilizes a different linearization than the one commonly used for nonlinear problems. The other adjoint is based on the property that our iterative multi-discretization Galerkin finite element method is a consistent discretization of the analytic iterative method.

We derive *a posteriori* error estimates to quantify various sources of error in a quantity of interest computed from our iterative finite element method. We first derive estimates for the case when the different components of the system are solved on the same spatial mesh, and then extend the analysis to include distinct meshes. The error estimator has terms indicating errors arising from discretization of each component, finite iteration, differences between the two different adjoints and projection. We demonstrate the accuracy of our method through a variety of numerical examples, starting from simple

linear problems and ending with the non-linear multi-scale Brusselator problem. We demonstrate how refining one or both meshes or increasing the number of iterations can decrease the specific error components arising from a specific source. Hence our error estimates are useful not only for computing the total error in a quantity of interest, but also applicable in guiding an adaptive refinement strategy.

9. Appendix

We prove the convergence of the iterative scheme in Alg. 2. We consider $u_i(t)$ as functions from the interval to a Banach space X, $u_i(t) : [t_{n-1}, t_n] \times X \to X$, where $X = L^2(\Omega)$, for $i = 1,2$. Let $f(u) : [t_{n-1}, t_n] \times X \times X \to X \times X$ be uniformly Lipschitz continuous with constant L, i.e. $\|f(u) - f(v)\|_{X\times X} \leq L\|u - v\|_{X\times X} \forall t$. Let $-A_i = -\nabla \cdot \epsilon_i \nabla$ be the infinitesimal generator of the C_0 semigroup $G_i(t), t \geq 0$, on X. For simplicity of notation, we denote $A_1 = A_2 = A$ and $G_1 = G_2 = G$. Then, based on the theory of semigroups [24], (6) and (7) on an interval $[t_{n-1}, t_n]$ are recast as,

$$u_1^{(mi)}(t) = G(t - t_{n-1})u_1^{(M_{n-1})} + \int_{t_{n-1}}^{t} G(t-s) f_1(u_1^{(mi)}, u_2^{(m-1)})\, ds \tag{58}$$

$$u_2^{(mi)}(t) = G(t - t_{n-1})u_2^{(M_{n-1})} + \int_{t_{n-1}}^{t} G(t-s) f_2(u_1^{(m)}, u_2^{(mi)})\, ds \tag{59}$$

Let M denote the bound on $\|G(t)\|$ on $[0, T]$. We then have,

Lemma 9.1. *With Assumptions A.1 and A.2, the integral equation*

$$\xi(t) = G(t - t_{n-1})\alpha + \int_{t_{n-1}}^{t} G(t-s) f_i(\xi, \beta)\, ds \tag{60}$$

admits a unique solution (ξ, β).

Proof. The proof follows arguments used for ordinary differential equations [14] and employs techniques from [24]. Set $\xi^{(0)} = \alpha$ and compute

$$\xi^{(j)} = G(t - t_{n-1})\alpha + \int_{t_{n-1}}^{t} G(t-s) f_i(\xi^{(j-1)}, \beta)\, ds, \tag{61}$$

for $j = 1,2,\ldots..$ For $j = 1$ we have,

$$\begin{aligned}
\|\xi(t) - G(t - t_{n-1})\alpha\| &= \left\| \int_{t_{n-1}}^{t} G(t-s) f_i(\alpha, \beta)\, ds \right\|, \\
&= \left\| \int_{t_{n-1}}^{t} G(t-s)(f_i(\alpha, \beta) - f_i(0,0))\, ds \right\|, \quad \text{(since } f_i(0) = 0\text{)} \\
&\leq \Delta t_n M L \|(\alpha, \beta)\|.
\end{aligned}$$

Moreover, using a semigroup property (cf. page 5 in [24]),

$$\|G(t - t_{n-1})\alpha - \alpha\| = \left\| \int_0^{t - t_{n-1}} G(s) A\alpha\, ds \right\| \leq \Delta t_n M \|A\alpha\|.$$

Using the above results and the triangle inequality,

$$\|\xi^{(1)}(t) - \alpha\| \leq \|\xi^{(1)}(t) - G(t - t_{n-1})\alpha\| + \|G(t - t_{n-1})\alpha - \alpha\| \leq \Delta t_n M L c_1,$$

where $c_1 = \|(\alpha, \beta)\| + L^{-1}\|A\alpha\|$. Now we use induction argument, where our induction hypothesis is

$$\|\xi^{(j-1)}(t) - \xi^{(j-2)}(t)\| \leq c_1 (ML\Delta t_n)^{(j-1)} \tag{62}$$

Then, using the Lipschitz continuity of f and our induction hypothesis (62) we have,

$$\begin{aligned}\|\xi^{(j)}(t)-\xi^{(j-1)}(t)\| &= \int_{t_{n-1}}^{t} G(t-s)(f_i(\xi^{(j-1)},\beta)-f_i(\xi^{(j-2)},\beta))\,ds \\ &\le \Delta t_n ML\|\xi^{(j-1)}(t)-\xi^{(j-2)}(t)\| \\ &\le c_1(ML\Delta t_n)^j\end{aligned}$$

Now, if $ML\Delta t_n < 1$, then for $l > k > N$,

$$\|\xi^{(l)}(t)-\xi^{(k)}(t)\| \le \sum_{j=k+1}^{l} \|\xi^{(j)}(t)-\xi^{(j-1)}(t)\| \le \frac{c_1(ML\Delta t_n)^N}{1-ML\Delta t_n} \tag{63}$$

Thus, $\|\xi^{(l)}(t)-\xi^{(k)}(t)\| \to 0$ as $N \to \infty$. Hence, $\xi^{(l)}(t)$ is a Cauchy sequence in the Banach space X, and hence converges to an element in X. We pass to the limit in (61), so that this limit satisfies (60). □

Now we use this lemma to prove the convergence of Alg. 2.

Theorem 9.1. *With Assumptions A.1 and A.2, there exists $t_n > t_{n-1}$ such that the sequence of functions $\{u_1^{(m)}\}$ and $\{u_2^{(m)}\}$ as defined in Alg. 2 converges to the exact solution of* (2) *on the time interval $I_n = [t_{n-1}, t_n]$.*

Proof. The existence of the sequences $\{u_1^{(m)}\}$ and $\{u_2^{(m)}\}$ are established by repeated application of Lemma 9.1. For $m = 1$, we set $\alpha = u_1(t_{n-1})$ and $\beta = u_2^{(0)}(t_{n-1})$. Then, by Lemma 9.1, there exists a solution $(u_1^{(1)}, u_2^{(0)})$ to the integral equation governing $u_1^{(1)}$. We obtain a similar result for $u_2^{(1)}$ by setting $\alpha = u_2(t_{n-1})$ and $\beta = u_1^{(1)}(t_{n-1})$. Hence, repeated application of this lemma shows the existence of the sequences $\{u_1^{(m)}\}$ and $\{u_2^{(m)}\}$. Moreover, from the proof of Lemma 9.1 we have,

$$\|u_2^{(1)}(t)-u_2^{(0)}(t)\| = \|u_2^{(1)}(t)-u_2(t_{n-1})\| \le c_1 ML\Delta t_n$$

Thus,

$$\begin{aligned}\|u_1^{(2)}(t)-u_1^{(1)}(t)\| &\le \int_{t_{n-1}}^{t} \|G(t-s)(f_1(u_1^{(2)},u_2^{(1)})-f_1(u_1^{(1)},u_2^{(1)}))\| + \|G(t-s)(f_1(u_1^{(1)},u_2^{(1)})-f_1(u_1^{(1)},u_2^{(0)}))\|\,ds \\ &\le ML\int_{t_{n-1}}^{t} \|u_1^{(2)}-u_1^{(1)}\|\,ds + c_1 ML(t-t_{n-1})^2\end{aligned}$$

Setting $\tau_n = ML\Delta t_n \exp(ML\Delta t_n)$, we apply Gronwall's inequality,

$$\|u_1^{(2)}(t)-u_1^{(1)}(t)\| \le c_1 ML(t-t_{n-1})^2 \exp(ML\Delta t_n) \le \frac{c_1\tau_n^2}{ML\exp(ML\Delta t_n)} \tag{64}$$

Similarly,

$$\|u_2^{(2)}(t)-u_2^{(1)}(t)\| \le \frac{c_1\tau_n^2}{ML\exp(ML\Delta t_n)} \tag{65}$$

Now we use induction, where our induction hypothesis is,

$$\|u_1^{(m-1)}(t)-u_1^{(m-2)}(t)\| \le c_n\tau_n^{m-1} \tag{66}$$

and

$$\|u_2^{(m-1)}(t)-u_2^{(m-2)}(t)\| \le c_n\tau_n^{m-1} \tag{67}$$

where $c_n = \frac{c_1}{ML\exp(ML\Delta t_n)}$. We have, by our induction hypothesis and Gronwall's inequality,

$$\begin{aligned}
\|u_1^{(m)}(t) - u_1^{(m-1)}(t)\| &\leq \int_{t_{n-1}}^{t} \|G(t-s)(f_1(u_1^{(m)}, u_2^{(m-1)}) - f_1(u_1^{(m-1)}, u_2^{(m-1)}))\| \\
&\quad + \|G(t-s)(f_1(u_1^{(m-1)}, u_2^{(m-1)}) - f_1(u_1^{(m-1)}, u_2^{(m-2)}))\|\, ds \\
&\leq ML\int_{t_{n-1}}^{t} \|u_1^{(m)} - u_1^{(m-1)}\|\, ds + ML\Delta t_n \|u_2^{(m-1)}(t) - u_2^{(m-2)}(t)\| \\
&\leq ML\int_{t_{n-1}}^{t} \|u_1^{(m)} - u_1^{(m-1)}\|\, ds + c_n ML\Delta t_n \tau_n^{m-1} \\
&\leq ML\Delta t_n c_n \tau_n^{m-1} \exp(ML\Delta t_n) \\
&= c_n \tau_n^m
\end{aligned} \tag{68}$$

For $\tau_n < 1$, and $l > k > N$,

$$\|u_1^{(l)}(t) - u_1^{(k)}(t)\| \leq \sum_{m=k+1}^{l} \|u_1^{(m)}(t) - u_1^{(m-1)}(t)\| \tag{69}$$

$$\leq \sum_{m=N}^{\infty} \|u_1^{(m)}(t) - u_1^{(m-1)}(t)\| \tag{70}$$

$$\leq \frac{c_n \tau_n^N}{1-\tau_n} \tag{71}$$

By enforcing $\tau_n < 1$, we get that $u_1^{(m)}$ is a Cauchy sequence that converges to an element in X. This is also true for $u_2^{(m)}$. We pass to the limit in (58), so that it converges to the solution of the implicit equation. □

[1] G. Adomian, The diffusion-Brusselator equation, Computers and Mathematics with Applications 29 (1995) 1–3.

[2] M. Ainsworth, T. Oden, *A posteriori* error estimation in finite element analysis, John Wiley-Teubner, 2000.

[3] W. Bangerth, R. Rannacher, Adaptive Finite Element Methods for Differential Equations, Birkhauser Verlag, 2003.

[4] T.J. Barth, A-Posteriori Error Estimation and Mesh Adaptivity for Finite Volume and Finite Element Methods, volume 41 of *Lecture Notes in Computational Science and Engineering*, Springer, New York, 2004.

[5] R. Becker, R. Rannacher, An optimal control approach to *a posteriori* error estimation in finite eement methods, Acta Numerica (2001) 1–102.

[6] V. Carey, D. Estep, S. Tavener, *A posteriori* analysis and adaptive error control for operator decomposition solution of elliptic systems I: Triangular systems, SIAM J. Numer. Anal. 47 (2009) 740–761.

[7] V. Carey, D. Estep, S. Tavener, *A posteriori* analysis and adaptive error control for multiscale operator decomposition solution of elliptic systems II: Fully coupled systems, Int. J. Numer. Meth. Engr. (2012, in revision).

[8] K. Eriksson, D. Estep, P. Hansbo, C. Johnson, Introduction to adaptive methods for differential equations, in: Acta Numerica, 1995, Acta Numerica, Cambridge Univ. Press, Cambridge, 1995, pp. 105–158.

[9] K. Eriksson, D. Estep, P. Hansbo, C. Johnson, Computational Differential Equations, Cambridge University Press, Cambridge, 1996.

[10] D. Estep, *A posteriori* error bounds and global error control for approximation of ordinary differential equations, SIAM J. Numer. Anal. 32 (1995) 1–48.

[11] D. Estep, Error estimation for multiscale operator decomposition for multiphysics problems, in: J. Fish (Ed.), Bridging the Scales in Science and Engineering, Oxford University Press, 2008.

[12] D. Estep, V. Carey, V. Ginting, S. Tavener, T. Wildey, *A posteriori error* analysis of multiscale operator decomposition methods for multiphysics models, J. Phys. Conf. Series 125 (2008) 012075.

[13] D. Estep, V. Ginting, D. Ropp, J. Shadid, S. Tavener, An *a posteriori-a priori* analysis of multiscale operator splitting, SIAM J. Numer. Anal. 46 (2008) 1116–1146.

[14] D. Estep, V. Ginting, S. Tavener, *A posteriori* analysis of a multirate numerical method for ordinary differential equations, CMAME 223-224 (2012) 10–227.

[15] D. Estep, M.G. Larson, R.D. Williams, Estimating the error of numerical solutions of systems of reaction-diffusion equations, Mem. Amer. Math. Soc. 146 (2000) viii+109.

[16] D. Estep, S. Tavener, T. Wildey, *A posteriori* analysis and improved accuracy for an operator decomposition solution of a conjugate heat transfer problem, SIAM J. Numer. Anal. 46 (2008) 2068–2089.

[17] D. Estep, S. Tavener, T. Wildey, *A posteriori* error analysis for a transient conjugate heat transfer problems, Finite Elem. Anal. Design 45 (2009) 263–271.

[18] D. Estep, S. Tavener, T. Wildey, *A posteriori* error estimation and adaptive mesh refinement for a multi-discretization operator decomposition approach to fluid-solid heat transfer, J. Comput. Phys. 229 (2010) 4143–4158.

[19] M. Giles, E. Süli, Adjoint methods for PDEs: A posteriori error analysis and postprocessing by duality, Acta Numerica (2002) 145–236.

[20] O.A. Ladyženskaja, V.A. Solonnikov, N.N. Ural'ceva, Linear and quasilinear equations of parabolic type, Translated from the Russian by S. Smith. Translations of Mathematical Monographs, Vol. 23, American Mathematical Society, Providence, R.I., 1968.

[21] M.G. Larson, F. Bengzon, Adaptive finite element approximation of multiphysics problems, Communications in Numerical Methods in Engineering 24 (2008) 505–521.

[22] M.G. Larson, A. Maylqvist, Goal oriented adaptivity for coupled flow and transport problems with applications in oil reservoir simulations, Computer Methods in Applied Mechanics and Engineering 196 (2007) 3546 – 3561.

[23] A. Logg, Multi-adaptive time integration, Appl. Numer. Math. 48 (2004) 339–354.

[24] A. Pazy, Semigroups of linear operators and applications to partial differential equations, Applied Mathematical Sciences, Springer-Verlag, 1983.

[25] I. Prigogine, R. Lefever, Symmetry breaking instabilities in dissipative systems, J. Chem. Phys. 48 (1968) 1695–1700.